République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et Systèmes Distribués (R.S.D)

Thème

Déploiement efficace des RSUs dans les VANETs en utilisant des approches métaheuristiques

Réalisé par :

- **TLEMSANI Imad Eddine**

Présenté le 2 Juillet 2018 devant le Jurycomposé de :

Président :	**Mr BENMAMMAR Badr**
Encadrant :	**Mr LEHSAINI Mohamed**
Examinateur :	**Mme AMRAOUI Asma**

Année universitaire :2017-2018

Remerciements

Ce travail de mémoire de fin d'études s'est accompli grâce à ***Dieu*** *vers lequel vont*
toutes nos louanges.
*Jeremercievivement notre encadreur, Mr****LEHSAINI Mohamed*** *de m'avoir accompagné dans la réalisation de ce travail.*
Son soutien, sa confiance et sa disponibilité m'a permis de travailler sereinement
durant toute la période de préparation de cette mémoire.

Je tienségalement à remercier monCo-encadreur Mr ***NEBBOU Tawfiq***
pour sadisponibilité et ses judicieux conseils.
Aussi à tous ceux et celles qui de près ou de loin qu'ils m'ont accompagné et soutenu
durant mon parcours de mes études, je pense à tous les enseignants de l'universitéAbou BakrBelkaid faculté des sciences.
Enfin, mes remerciements vont également aux membres du jury pour leurs apportsqu'ils ont portés à mon humble PFE, en acceptant de l'examiner et de l'enrichir parleurs remarques.

Merci

Dédicaces

À
Ma chère mère,
Ma chère Vie,
Ma Tante,
Mes Frères,
Mes Cousins,
Ma Famille,
Mes Amis,

Tous ceux qui me sont chers.

Imad

Table des matières

Liste des figures

Liste des tableaux

Introduction Générale

Introduction générale

Les réseaux Adhoc véhicules (VANETs) Les réseaux VANETs sont un type particulier de réseaux sans fil dans lequel lesnœuds sont des véhicules en mouvement sur les routes. Ce type de réseaux est une version spéciale des réseaux mobiles ad-hoc (MANETs) avec des spécificités supplémentaires. Ils permettent la communication et l'échange d'informations entre lesusagers de la route.

Les réseaux VANETs sont caractérisés par leur grande mobilitégrâce à la mobilité des nœuds à desgrandes vitesses. Ils sont déployés dans des routes ainsi que dans les autoroutes.Il existe trois types de communication dans les VANETs à savoir V2I (Vehicle-to-Infrastructure), où les véhicules dotés d'OBU[1]communiquentdirectement avec les RSUs[2], V2V (Vehicle-to-Vehicle), où les véhicules communiquententre eux sans passer par les RSUs et V2X qui combinent les deux types de communications.

Les RSUs sont généralement placées dans les intersections surtout dans les environnements urbains et jouent un grand rôle dans la dissémination des informations routières. Cependant, leur coût est plus ou moins élevé. De ce fait, il est judicieux de placer un nombre minimal de RSUs tout en garantissant un routage de données avec une qualité acceptable.

Ce travail est une amélioration du PFE [1] dans lequel les auteurs ont supposés qu'au niveau de chaque intersection une RSU est placée. C'est vrai les résultats obtenus par cette configuration sont bons mais le coût de la mise en place de ces RSUs rend cette configuration non rentable. Dans cette optique, nous avons visé à minimiser le nombre de RSUs placées dans un environnement urbain et nous avons pris comme exemple d'environnement la localité d'Abou Tachfine de la wilaya de Tlemcen là où il existe 80 intersections. Nous avons choisi comme critère de performance le taux de couverture de la zone prise en considération. Nous avons constaté que le nombre de configurations est très grand quand le nombre des RSUs est petit. A cet effet, l'exploration de toutes les configurations demande un temps de calcul non raisonnable. D'où, nous avons fait recours à deux métaheuristiques différentes pour alléger ce temps de calcul : les algorithmes génétiques et le recuit simulé. Par ailleurs, pour accélérer la convergence de ces métaheuristiques et trouver de bons résultats, nous avons proposé une amélioration du contexte d'exécution de ces deux métaheuristiques. Dans ce cadre, nous avons exploité chacune de ces deux métaheuristiques dans un contexte basique et dans un contexte amélioré.

Pour ce faire, notre mémoire est organisé comme suit :

Le chapitre 1 est un survol sur les réseaux véhiculaires AdHoc (VANETs) avec définitions, description des entités et présentation de leurs caractéristiques.

[1] OBU : OnBoard Unit
[2] RSU : RoadSide Unit

Le chapitre2 est consacré à la présentation des deux métaheuristiques utilisées pour traiter notre problématique à savoir les algorithmes génétiques et le recuit simulé.

Le chapitre 3 présente en détails les différentes solutions proposées pour traiter la problématique abordée dans le cadre de notre projet de fin d'études.

Enfin, une conclusion générale et des perspectives clôturent notre mémoire.

Chapitre I
Généralités sur les réseaux VANETs

Chapitre 1 Généralités sur les réseaux VANETs

1.1 Introduction

VANET (**V**ehicular **A**d hoc **NET**work) est une nouvelle technologie qui utilise les véhicules comme des nœuds pour créer un réseau mobile. Ces véhicules sont équipés d'interfaces sans fil leur permettant de communiquer entre eux. En effet, les VANETs peuvent êtreutilisés pour étendre la portée des informations de sécurité (messages d'alerte,informations sur les anomalies, etc.) ou d'autres types d'applications (multimédia, …).

Les réseaux véhiculaires sont une projection des systèmes de transports intelligents(Intelligent Transportation Systems - ITS). Les véhicules peuvent communiquer les unsavec les autres par l'intermédiaire aussi bien qu'avec les équipements de la route parl'intermédiaire de la communication d'équipement-à-Véhicule (Roadside-to-Vehicle).

Dans ce type de réseaux, tous les véhiculesjouent le rôle comme routeurs en permettant de relayer les informations d'un véhicule à un autre jusqu'à l'arrivée au véhicule destinataire. La portée de communication dans les VANETs peut atteindre 300m. Comme les véhicules peuvent se trouver hors de la portée du signal et abandonnent le réseau, d'autres véhicules peuvent rejoindre le réseau véhiculaire, reliant les véhicules les uns aux autres de sorte qu'un Internet mobile soit créé. Nous estimons que le premier système qui a intégré cette nouvelle technologie est le réseau véhiculaire de la police qui permet aux véhicules qui le compose de communiquer les uns avec les autres pour des besoins de sécurité.

Les réseaux ad-hoc véhiculaires sont censés mettre en œuvre une variété de technologies sans fil telles que DSRC (Dedicated Short Range Communications) [2]qui est un type de Wi-Fi. Les autres technologies sans fil sont les suivantes : cellulaire, satellite et WiMAX. Les VANETs peuvent être considérés comme composante du Transport Intelligent Systèmes (ITS)[3].

Ce chapitre est consacré à la présentation des réseaux ad hoc sans fil. Nous mettonsl'accent sur les réseaux VANETs avec description de la norme IEEE 802.11p [4]et sonmécanisme d'accès au canal.

1.2 Réseaux VANETs

Les réseaux VANETs constituent une nouvelleforme de réseaux ad hoc mobiles (MANETs). Ils permettent d'établir descommunications entre véhicules ou bien avec uneinfrastructure située aux bords de routes (RSU : RoadSideUnits). Comparativement à unréseau ad hoc classique, les réseaux VANETs sont caractérisés par une forte mobilitédes nœuds rendant la topologie du réseau fortement dynamique.

Le réseau ad-hoc véhiculaire est une application prometteuse du réseau ad hoc sansfil. Pour sa mise en place, certains équipements électroniques doiventêtre installés au sein de véhicules (Figure I.1), tels que : les dispositifs de perception del'environnement (radars, caméras), un système de localisation GPS, et bien sûr uneplateforme de traitement [5].

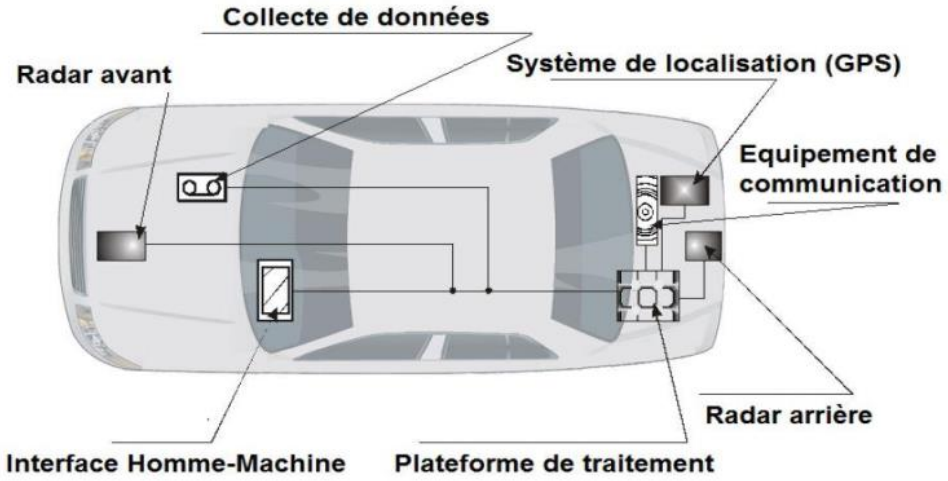


Figure I.1: Les éléments constituant le véhicule intelligent [5]

Un exemple de réseaux VANETs est illustré dans la Figure I.2.

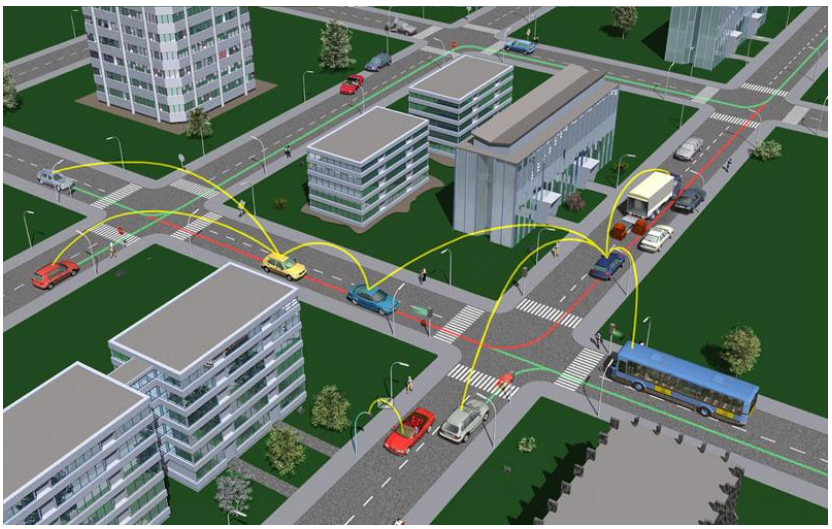


Figure I.2: Exemple de réseaux VANETs [5]

1.2.1 Les composants d'un réseau VANET

Les principaux composants nécessaires pour garantir les communications d'un réseau VANET sont les suivantes :

a) OBU (OnBoard Unit)

L'OBU (Figure I.3) est un dispositif électronique installé au seindes véhicules intelligents, contient un ensemble de composants logiciels pour calculer et afficher toutes les informations nécessaires.

Figure I.3: OnBoard Unit

b) RSU (RoadSide Unit)

Une RSU (Road Side Unit) (Figure I.4) est un dispositif installé au bord de la route joue le rôle d'un point d'accès afin d'assurer les communications V2I.

Figure I.4: RoadSide Unit

1.2.2 Les services offerts par les réseaux VANETs

Il existe plusieurs services peuvent être offerts par les réseaux VANETs, ils sont classés comme suit :

a) Les services liés à la sécurité routière

Ces services concernent les applications ayant un impact direct sur la sécurité despersonnes et des biens. Ils se basent sur la détection de l'environnementproche au moyen de capteurs (par exemple : les radars et les caméras) installés auniveau des véhicules ou bien au centre de contrôle, ainsi que la diffusion de messagesfournissant des informations sur l'état du réseau routier (trafic, travaux, météo), ourappelant au conducteur les limitations de vitesse, les distances de sécurité ou qu'ils'approche d'une intersection.

b) Services liés au confort

En plus des services liés à la sécurité routière, d'autres services assurent le confort des passagers ; ces services peuvent être : La communication multimédia, les jeux enréseau, la messagerie instantanée, l'accès à Internet, les paiements automatiques et ladiffusion d'informations utiles.

1.2.3 Les modes de communication dans les réseaux VANETs

On peut distinguer trois modes de communication, les communications Véhicule-à-Véhicule (V2V), les communications Véhicule -à- Infrastructure (V2I) et les communications qui combinent les deux modes (V2X) (Figure I.5).

a) Mode de communication Véhicule-à-Véhicule (V2V)

Ce mode de communication est basé sur une communication inter-véhicules sansutiliser une infrastructure. En effet, un véhicule peut communiquer directementavec un autre véhicule s'il se situe dans sa zone de couverture, ou bien par le biais d'unprotocole multi-sauts qui se charge de transmettre les messages de bout en bout enutilisant les nœuds voisins qui les séparent comme des relais. Dans ce mode, lesinterfaces de communication utilisées sont caractérisées par un grand débit detransmission et une petite latence.

Les communications V2V peuvent être efficaces pour le transfert des informations desécurité routière, mais elles ne garantissent pas une connectivité permanente entreles véhicules, ce qui la rend moins efficace pour les applications avec une grandequantité de données comme le multimédia en particulier quand le trafic véhiculaire est faible.

b) Mode de communication de Véhicule à Infrastructure (V2I)

Ce mode de communication utilise des infrastructures (Road Side Units)déployées aux bords des routes pour fournir des services comme l'accès à Internet, lescommunications de voiture-à-garage de réparation pour le diagnostic distant, etautres.

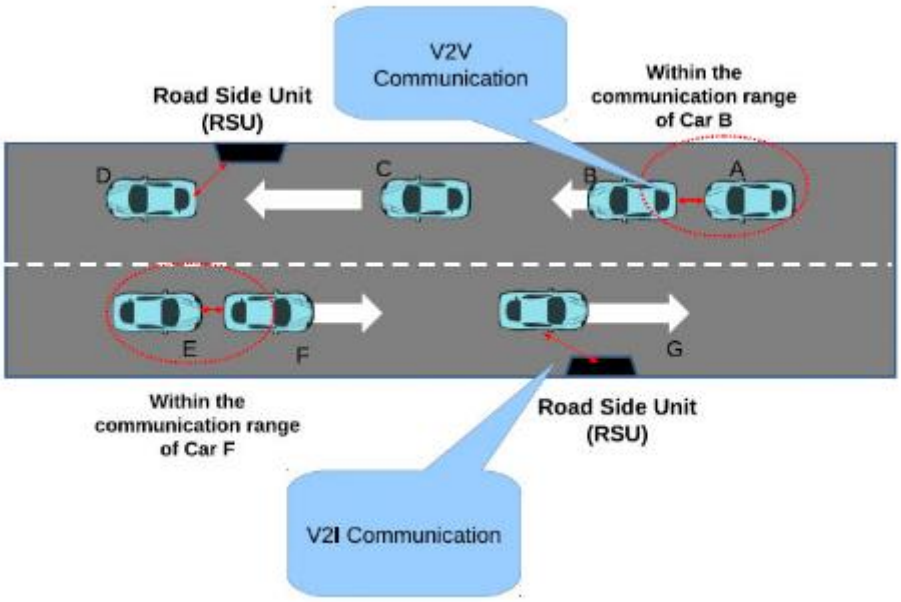


Figure I.5: Les modes de communication dans les VANETs [6]

c) Mode de communication hybride (V2X)

Les communications entre les véhicules s'effectuent soit en mode V2V si le véhicule destinataire se trouve dans la gamme de transmission ou bien en mode V2I si la destination se trouve hors la gamme de transmission. Ces deux modes sont généralement combiné pour donner naissance un mode de communication hybride comme montre la Figure I.6.

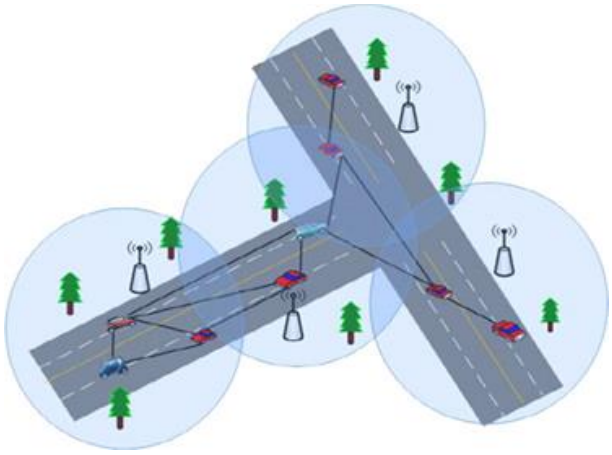


Figure I.6: Mode communication hybride

1.2.4 Les caractéristiques des VANETs

Les réseaux VANETs ont des caractéristiques spécifiques qui les distinguent deréseaux ad hoc mobiles. Ces caractéristiques doivent être prises en compte lors deconception des protocoles pour les VANETs. Dans ce que suit, nous présentons quelques propriétés et contraintes sous-jacentes à ce type de réseaux :

- **Le modèle de mobilité :** Le modèle de mobilité est généralement différent dans les réseaux VANET par rapport aux autres réseaux sans fil ad hoc, car plusieurs facteurs peuvent affecter la mobilité comme le type des routes (autoroutes, villes, intersections, panneaux de signalisation,…). En outre, la mobilité dans les VANETs est liée directement au comportement des conducteurs et leurs réactions face à différentes situations (accident, embouteillage, signalisation,…).
- **La capacité d'énergie et stockage :** Les éléments des réseaux VANETs disposent suffisamment d'énergie pour alimenter les différents équipements électroniques, ce n'est pas le cas pour les réseaux MANETs ou les réseaux de capteurs. Donc, les nœuds sont censés avoir une grande capacité detraitement et de stockage de données.
- **La sécurité et l'anonymat :** l'importance des informations échangéesvia les communications véhiculaires rend l'opération de sécurisation de cesréseaux nécessaire et importante.
- **La topologie et la connectivité :**un véhicule peut rejoindre ou quitter ungroupe de véhicules en un temps très court, ce qui nous mène à avoir unetopologie très dynamique, et imprévisible avec une connectivité peu garantie et dépendante de la capacité du trafic routier.
- **Topologie dynamique** :Si nous supposons que deux véhicules s'éloignent l'un de l'autre avec une vitesse de 60 km/h et que la portée de transmission est d'environ 250 m, le lien entre ces deux véhicules ne durera que 5 secondes. Ceci définit sa topologie hautement dynamique.
- **Déconnexion fréquente**:La caractéristique ci-dessus nécessite que toutes les 5 secondesles nœuds aient besoin d'un autre lien avec le véhicule à proximité pour maintenir une connectivité transparente. Mais en cas d'une telle défaillance, en particulier dans le cas d'une zone à faible densité de véhicules, une perturbation fréquente de la connectivité du réseau se produira. Ces problèmes sont parfois résolus par le déploiement des nœuds relais sur la route.
- **Modélisation et prévision de la mobilité** :Les caractéristiques citées en haut pour la connectivité nécessitent donc la connaissance des positions des nœuds et de leurs mouvements, ce qui est très difficile à prédire compte tenu de la nature et de la structure du mouvement de chaque véhicule. Néanmoins, un modèle de mobilité et une prédiction des nœuds basés sur l'étude du modèle de route prédéfinie et de la vitesse du véhicule sont d'une importanceprimordiale pour une conception de réseau efficace.
- **Contraintes de délai:**L'aspect de sécurité (tels que les accidents, les événements de freinage,…) des applications VANET garantit l'envoi des messages (alertes) aux véhicules concernés. Il ne peut tout simplement pas compromettre avec un retard de données dur à cet égard. Par conséquent, le débit de données n'est pas aussi important qu'un problème que les contraintes de retardement.
- **Interaction avec les capteurs embarqués :**Ces capteurs aident à fournir l'emplacement des véhicules et leur nature de mouvement (tel que GPS) et qui sont utilisés pour établir des liens de communication entre les véhicules pour un routage efficaces.

- **Capacité de calcul:**En effet, les véhicules peuvent offrir des capacités significatives de calcul, de communication et de perception de l'environnement.
- **Taille de réseau potentiellement non limitée:**LesVANETspeuvent impliquer les véhicules dans une ville, plusieurs villes ou même un pays. Ainsi, il est nécessaire de rendre les protocoles des VANETs évolutifs pour être pratiques.
- **Destinataire anonyme**:La plupart des applications dans les VANETs nécessitent l'identification des véhicules dans une certaine région, cela peut aider à protéger la confidentialité des conducteurs dans les VANETs.
- **Échange de données sensibles au temps:**La plupart des applications liées à la sécurité nécessitent une transmission de paquets de données en temps réel. Ainsi, le processus de sécurité ne doit pas nuire les performances réseau des VANETs.
- **Soutien potentiel par l'infrastructure :**Contrairement aux MANETs, les VANETs peuvent réellement communiquer avec les infrastructures. Cette propriété doit être prise en compte pour rendre les protocoles de routage efficace.

1.3 Les Normes et les Standards

Il existe plusieurs normes qui se rapportent à l'accès sans fildans les environnements véhiculaires.

1.3.1 IEEE 802.11p – WAVE

IEEE 802.11p [4]est une modificationapprouvée de la norme IEEE 802.11 pour ajouter leWAVE (Wireless Access forVehicular Environments). Pour prendre en charge les applications de systèmes detransport intelligents (ITS), elle impose des améliorations à la norme IEEE 802.11.

Selon la définition de l'IEEE, l'accès sans fil dans les environnements véhiculaires (WAVE) IEEE 1609.x [7] est un mode de fonctionnement utilisé par IEEE Std802.11 TM devices dans des environnements où les propriétés de la couche physiquechangent rapidement et où des échanges de communications de très courte durée sont nécessaires.

Les couches de la pile protocolaire de WAVE se situant entre la couche liaison dedonnées et la couche application représentent l'architecture de WAVE que le groupeIEEE1609 a standardisée sous quatre catégories comme suit :

- IEEE 1609.1 – WAVE *Resource Manager* : pour la gestion des ressources.
- IEEE 1609.2 – WAVE *Security Services for Applications and Management Messages* :pour la sécurisation des messages.
- IEEE 1609.3 – WAVE *Networking Services* : pour les services de niveau réseau ettransport incluant l'adressage, le routage.
- IEEE 1609.4 - WAVE *Multi-Channel Operation*: pour la coordination et la gestiondes sept canaux DSRC.

Le standard IEEE 1609.1 se positionne au niveau de la couche application et définit les formats des messages utilisés par la couche application, et définit un gestionnaire de ressources qui autorise les applications del'équipement de bord de route (RSU) à communiquer avec les OBU des véhicules se trouvant dans sa gamme de transmission.

1.3.2 DSRC (Dedicated Short-Range Communication)

La création de la première génération de systèmes detransport intelligents (ITS), dont l'objectif principal est d'améliorer la sécurité routière.A travers cette génération, c'était le premier effort visant à normaliser la communication des réseaux ad hocvéhiculaires. Après, DSRC (Communication à courte portée dédiée) a été indiquéecomme norme conçue pour l'usage automobile par le FCC[3]. La première génération du système de communication dédiée àcourte distance fonctionne à 915 MHz et a un taux de transmission de 0.5 Mbps.Ce projet a eu un succès limité et a été utilisé principalement pour des servicescommerciaux tels que le péage. Après 8 ans, ils ont pu allouer une bande passante de75 MHz dans la bande 5.9 GHz pour la deuxième génération de DSRC [2].Cette norme IEEE permet d'avoir des canaux de communication sans fil à un sens ouà double sens sur un rayon de 1000m avec un débit allant jusqu'à 27Mbps. Ellefonctionne sur une bande de fréquences autorisée, par contre, ces bandes ne sont pas compatiblesd'un pays à un autre.

DSRC est active dans une bande de fréquence des 5.9GHz (Europe et Etats-Unis) ou5.8GHz (Japon) et sur une largeur de bande de 75 MHz. Elle est segmentée en 7canaux de 10 MHz chacun avec les premiers 5Mhz utilisée comme intervalle de garde. L'ensemble des canaux se répartissent en un canal de contrôle (CCH) et 6 canaux de service (SCH). Les numéros de canaux sont déterminés par leur décalagede la fréquence centrale 5.000 GHz avec des unités de 5 Mhz (les premiers 10 Mhzsont répartis de 5.855 à 5.865 GHz avec une fréquence centrale de 5.860 GHz qui est860 Mhz au-dessus de la ligne de base, c'est un décalage de 172 unités de 5 Mhz d'oùle numéro 172). Les canaux 174 et 176 peuvent être combinés afin de former le canal175 de 20 Mhz, tel est le cas aussi pour les canaux 180 et 182. Le canal de contrôleest réservé à la transmission des messages de gestion du réseau et aux messages detrès haute priorité tels que les messages critiques liés à la sécurité routière. Les sixautres canaux sont quant à eux dédiés à la transmission des données des différentsservices annoncés sur le canal de contrôle. Le spectre DSRC est partagé entre les OBU et les RSU dans un espace donné. Avec cepartage une interférence est possible entre un nœud qui émet et un autre qui écoute.Deux types d'interférences sont identifiées ; interférence co-canal (si les deux nœudsutilisent le même canal), l'interférence entre deux canaux adjacents.

Le spectre DSRC 5.9 GHz est composé de six canaux de service (SCH) et d'un canal de commande (CCH) comme montre la Figure I.7.

[3] FCC : Federal CommunicationsCommission

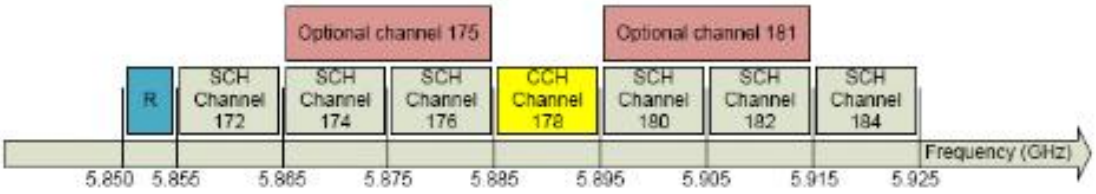


Figure I.7: Canaux alloués par DSRC [2]

Ces canaux sont spécifiés par la norme DSRC. En utilisant ces canaux à 10 MHz, desdébits de données de 3, 6, 9, 12, 18, 24 et 27 Mbps sont autorisés, y compris unpréambule de 3 Mbps. Le schéma de modulation utilisé par DSRC est le multiplexagepar répartition orthogonale de la fréquence (OFDM). Le canal de contrôle CCH estdédié aux trames de diffusion pour les applications de sécurité, les annonces deservice et les messages de véhicule à véhicule. Les autres canaux, les canaux deservice SCH, prennent en charge à la fois la sécurité et les applications orientéesutilisateur, et pourraient également être utilisés pour diffuser des messages.

1.4 Conclusion

Dans ce chapitre, nous avons présenté les réseaux véhiculaires ad hoc (VANETs) qui ne sont qu'un sous-groupe des réseauxMANETs. Ainsi, nous avons décrit les composants des réseaux VANETs, leur technologie de communications, les principales caractéristiqueset leurs domaines d'applications.

Dans le chapitre suivant, nous présenterons les métaheuristiques pour résoudre notre problématique dans le cadre de notre projet de fin d'études.

Chapitre II
Généralités sur les TechniquesD'Optimisation

Chapitre 2
Généralités sur les techniques d'optimisation

1.1 Introduction

Il existe de nombreux problèmes intéressants pour desapplications pratiques où l'application d'une méthode déterministe pour les résoudre est quasiment impossible en terme de temps de calcul surtout lorsque l'espace de solutions réalisables est très grand. Il s'agit bien dans ce type d'applications d'un problème d'optimisation combinatoire, parmi ces applications, nous pouvons citer :

- Recherche de chemin dans un réseau dense,
- Conception de réseaux,
- Placement des bornes WIFI pour la couverture d'une grande région,
- Placement des RSUs pour la couverture d'une zone urbaine,
- …..

Pour ces problèmes, il n'existe aucun algorithme connuayant un temps d'exécution polynomial et les seuls algorithmes permettant d'obtenir une solutionoptimale ont une complexité asymptotique trèsélevée(complexité exponentielle $\theta(2^n)$ou factorielle $\theta(n!)$).

Dans le cadre de notre PFE nous visons à minimiser le nombre des RSUs placées dans les intersections pour garantir une couverture quasiment totale de la région tout en préservant des métriques de routage acceptables. Nous avons choisi la localité d'Abou Tachfine sise à Tlemcen dans laquelle il y a 80 intersections. Par exemple, si on veut placer 10 RSUs dans ces intersections, le nombre de solutions possibles à explorer est :

$$C_{10}^{80} = \frac{80!}{10! * 70!} = 16464921101.2 \text{ possibilités}$$

D'où le recours à une méthode approchée est devenu une nécessité. De ce fait, nous avons proposé de résoudre ce problème en utilisant deux métaheuristiques : les algorithmes génétiques et le recuit simulé. En plus, pour accélérer la convergence de ces deux métaheuristiques nous avons pris en compte les emplacements des RSUs là où il n'y a pas un grand chevauchement entre deux RSUs proches.

2.2 Optimisation Combinatoire

Les problèmes qui font recours à une optimisation combinatoire consistent d'être modélisés puis résolus en utilisant une méthode approchée.

2.2.1 Définition

Un problème d'optimisation noté $P(X, f)$ tel que X : l'ensemble des solutions réalisable ou admissible non-vide, f : la fonction Objectif qui associe un scalaire dans R à chaque élément x de X. $P(X, f)$ revient à trouver parmi les solutions réalisables une qui minimise ou maximise f.

$$x^* \in X/ \quad \forall x \in A \quad et \ A \subset X \qquad f(x) \geq f(x^*) \quad (Minimisation)$$

$$x^* \in X/ \quad \forall x \in A \quad et \ A \subset X \qquad f(x) \leq f(x^*) \quad (Maximisation)$$

Où A représente l'ensemble des solutions explorés.

Pour résoudre un problème d'optimisation combinatoire en vue de trouver une meilleure solution, il faut procéder comme suit :

- Modélisation du problème : Dans cette étape, on modélise le problème en le représentant dans un contexte analytique ou de simulation et on définit l'espace des solutions.
- Formulation Mathématique : Formaliser le problème et fixer une fonction Objectif.
- Résolution du modèle : en appliquant une méthode d'optimisation.
- Evaluation de la solution trouvée et si c'est nécessaire retourner la 1ière étape.

Le schéma représenté par la figure II.1 récapitule cette démarche.

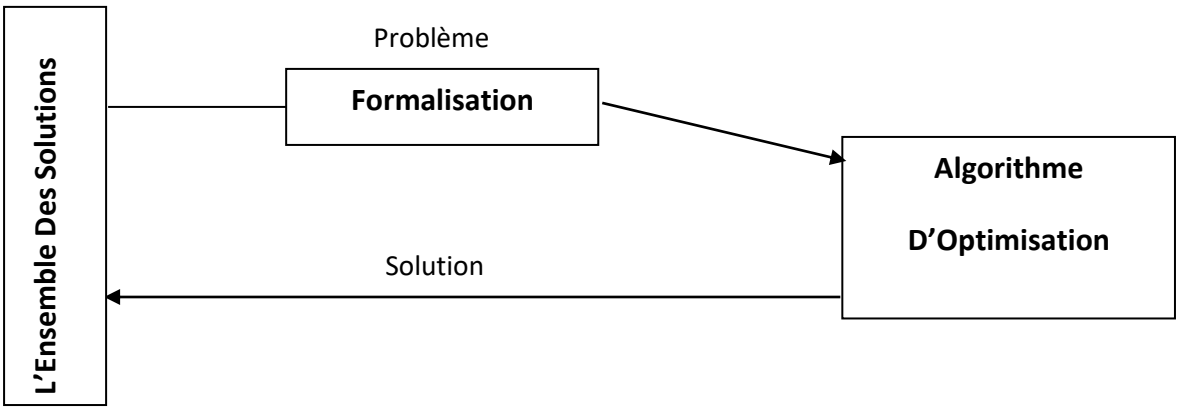


Figure II.1: Résolution d'un problème d'optimisation combinatoire

2.2.2 Contexte de résolution d'un problème d'optimisation combinatoire

La recherche d'une solution exacte optimale à l'aide desalgorithmes de complexité exponentielle ou factorielle est doncimpensable sauf si on dispose de beaucoup de ressources decalcul. Nous citons quelques exemples de ce type de problèmes :

- En 1997, le record pour le PVC (Problème du Voyageur de Commerce) pour une instance comptant 7397 villes, avait demandé 3 ans de temps CPU sur réseau de stationsSUN.
- En 2004, avec 24 978 villes, ce problème a demandé le même temps.

La théorie des problèmes NP-difficiles porte sur une classe deproblèmes pour lesquels il n'existe aucun algorithme connu quia une complexité polynomiale. Pour ces problèmes difficiles, puisqu'il est impossible d'obtenirune solution dans un temps raisonnable, on fait recours à des algorithmes qui fournissent des solutions "pas trop mauvaises".

2.2.3Les approches de résolution d'un problème de combinaisoncombinatoire

On distingue deux approches pour résoudre un problème de combinaison combinatoire comme montre la figure II.2:

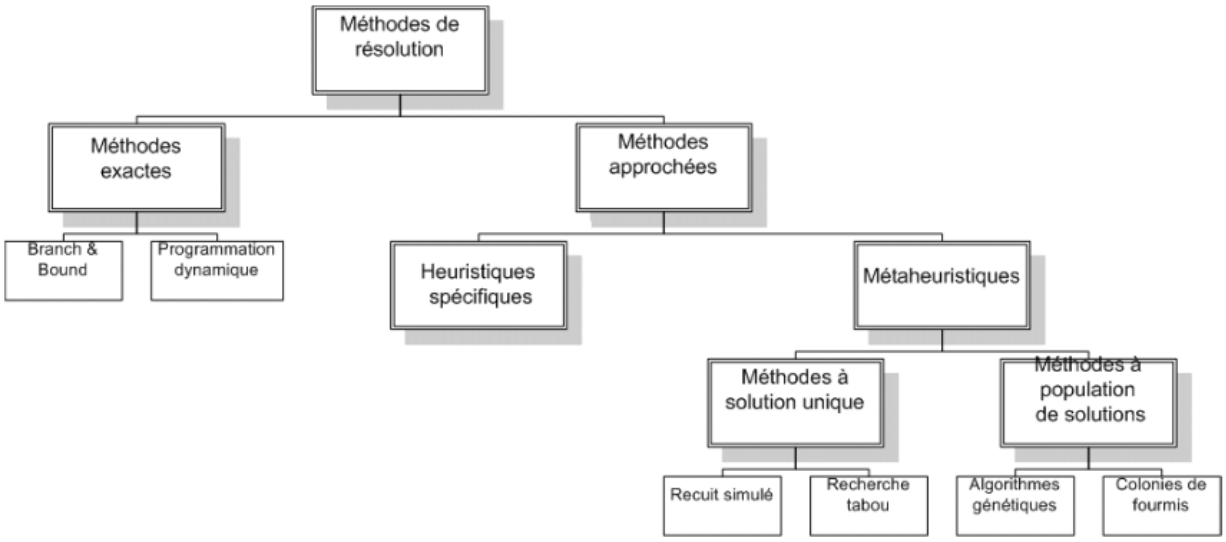


Figure II.2: Classification des méthodes d'optimisation combinatoire

a) Méthodes Exactes (Exhaustives)

Les méthodes exactes permettent d'explorer tout l'espace des solutions réalisables et par conséquent la solution optimale sera trouvée. Ces méthodes sont utilisées pour trouver au moins une solution optimale d'un problème. Les plus réussis dans la littérature appartiennent aux paradigmes de deux grandes classes [8] :

- **La programmation dynamique** : La programmation dynamique permet d'appréhender un problème de façon différente de celle que l'on pourrait imaginer au premier abord. Le concept de base est simple : une solution optimale est la somme de sous-problèmes résolus de façon optimale. Il faut donc diviser un problème donné en sous-problèmes et les résoudre un par un.

  La conception d'un algorithme de programmation dynamique peut être planifiée dans une séquence de quatre étapes :

  1. Caractériser la structure d'une solution optimale.
  2. Définir récursivement la valeur d'une solution optimale.
  3. Calculer la valeur d'une solution optimale en remontant progressivementjusqu'à l'énoncé du problème initial.
  4. Construire une solution optimale pour les informations calculées.

- **Les méthodes de recherche arborescente (Branch & bound)** : La méthode de branch and bound consiste à énumérer ces solutions d'une manière intelligente en ce sens que, en utilisantcertaines propriétés du problème en question. Cette technique arrive à éliminer des solutions partielles qui ne mènent pas à la solution que l'on recherche. De ce fait, on arrive souvent à obtenir la solution recherchée en des temps raisonnables. Bien entendu, dans le pire cas, on retombe toujours sur l'élimination explicite de toutes les solutions du problème.
  Pour ce faire, cette méthode se dote d'une fonction qui permet de mettre une borne surcertaines solutions pour soit les exclure soit les maintenir comme des solutions potentielles. Bien entendu, la performance d'une méthode de branch and bound dépend, entre autres, de laqualité de cette fonction. Elle s'exécute selon le schéma algorithmique suivant :

Début

Placer le nœud début de longueur 0 dans une liste.

Répéter

Si la première branche contient le nœud recherché alors

Fin avec succès.

Sinon

- Supprimer la branche de la liste et former des branches nouvelles en étendant la branche supprimée d'une étape.

- Calculer les coûts cumulés des branches et les ajouter dans la liste de telle sorte que la liste soit triée en ordre croissant.

Jusqu'à (liste vide ou nœud recherché trouvé)

Fin

Les méthodes exactes ne sont efficaces que pour les instances deproblèmes de petite taille etperdent leurs performances lors au passage à l'échelle.

b) Méthodes Approchées (Approximatives)

Explore un sous-ensemble de l'espace des solutions réalisables. Parmi ces méthodes nous citons :

- **Les algorithmes d'approximation** : Permettent d'obtenir une solution pas trop éloignée de la solution optimale. On réserve le qualitatif «algorithme d'approximation" aux algorithmes pour lesquels on peut définir de façon précise lanotion " pas de trop éloignée de la solution optimale".
- **Les algorithmiques heuristiques** : Produisent une solution qui approxime la solution optimale. Il n'est pas nécessairement possible de borner de façon exacte la qualité de la

solution produite relativement à la solution optimale. Une heuristique pourra produire une solution qui n'est pasbonne ou ne produire aucune solution. Les heuristiques peuvent être classées en deux catégories :

- Les méthodes constructives : ces méthodes génèrent des solutions de façon incrémentale, donc à partir d'une solution initialement vide à laquelle sont ajoutes des éléments jusqu'à l'obtention d'une solution plus ou moins complète.
- Les méthodes de fouilles locales : ces méthodes partent d'une solution initialement complète et de façon répétitive tented'améliorer cette solution en explorant son voisinage immédiat.

- **Les algorithmes probabilistes** : Dans ces algorithmes, quand on est confronté à un choix : ce choix se fait aléatoirement.Ces algorithmes ont un caractère non déterministe.
- **Les métaheuristiques** : ce sont des stratégies de résolution de problème qui utilisent, qui coordonnent d'autres heuristiques pour obtenir une solution à un problème difficile. Une heuristique est conçue pour un problème précis alors que les métaheuristiques sont conçues pour s'appliquer à diversproblèmes.

2.3 Métaheuristiques

Les métaheuristiques se sont des méthodes inspirées de la nature, ce sont des heuristiques modernes dédiées à la résolution des problèmes et plus particulièrement aux problèmes d'optimisation, qui visent d'atteindre un optimum global généralement enfoui au milieu de nombreux optimaux locaux [9].

Les métaheuristiques se subdivisent en deux sous-classes comme montre la figure 9:

- Méthode à solution unique,
- Méthodes à population de solutions

Dans cette section, nous détaillons une méthode de chaque classe : Recuit simulé pour les méthodes à solution unique et les algorithmes génétiques pour les méthodes à population de solution

2.3.1 Recuit Simulé

La méthode du recuit simulé est une généralisation de la méthode Monte-Carlo; sonbut est de trouver une solution optimale pour un problème donné. Elle a été mise au point partrois chercheurs de la société IBM : S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, etindépendamment par V. Cerny en 1985 à partir de l'algorithme de Metropolis ; qui permet dedécrire l'évolution d'un système thermodynamique [10].

Recuit Simulé est une métaheuristique qui est connue par deux caractéristiques : la facilité de son implémentation, et sa rapidité d'exécution comparée à d'autres métaheuristiques décrites dans la littérature. En outre, elle fournit de bons résultats pour de nombreux problèmes d'optimisation combinatoire.

Nous pouvons la considérer comme une version étendue de la méthode de descente. Elle débutera avec une initialisation d'une population X et le paramètre de contrôle C. Puis, elle génèrede manière aléatoire une solution de voisinage X' ∈ N(X) et elle évalue. Puis elle compare ses performances à la solution courante. La population X' sera retenue s'il y a une amélioration ou s'il y a une dégradation minime de la solution courante selon une certaine probabilité.

Son exécution se fait selon le schéma algorithmique suivant :

```
Entrée : X0
Sortie : X qui minimise F

DEBUT
  X ← X0
  T ← T0
  while Non condition d'arrêt do
    m ← 0
    repeat
      Y ← voisin(X)
      dF ← F(Y) − F(X)
      if Accepte(dF, T) then
        X ← Y
      end if
      m ← m + 1
    until m = NT {nombre d'itérations maximum à la température T}
    T ← Decroissance(T)
  end while
FIN
```

La fonction d'acceptation ***Accepte(dF,T)***est comme suit :

```
Entrée : dF, T

Résultat : VRAI si la fonction F décroit
           VRAI en fonction d'une certaine probabilité si F croit
           FAUX dans les autres cas

DEBUT
  if dF < 0 then
    Retourner VRAI
  else
    A ← e^(-dF/T)
    if Alea(0,1) < A then
      Retourner VRAI
    else
      Retourner FAUX
    end if
  end if
FIN
```

L'algorithme de Recuit Simulé s'arrête quand la température devient nulle et la meilleure solution trouvée parmi les solutions explorées sera retournée comme résultat.

2.3.2 Les Algorithmes Génétiques

Les algorithmes génétiques sont des algorithmes d'optimisation stochastique fondés sur les phénomènes de la sélection naturelle et de la génétique. Leur fonctionnement est extrêmement simple. On partavec une population de solutions potentielles (individus ou chromosomes) initiales arbitrairement choisies. On évalue leur performance (fitness) relative et sur la base de cesperformance on crée une nouvelle population de solutionspotentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement et la mutation. On répète ce cycle jusqu'à ce que l'on trouve une solution satisfaisante.

a) Présentation des algorithmes génétiques

Un algorithme génétique est défini par :

- Individu/chromosome : une solution potentielle du problème ;
- Population : un ensemble de chromosomes ou de points de l'espace de recherche ;
- Environnement : l'espace de recherche ;
- Fonction de fitness : la fonction Objectif que nous cherchons à maximiser ou à minimiser.

Nous présentons dans ce que suit quelques notions liées aux algorithmes génétiques.

- **Définition 1 (Chromosome ou Individu (codage binaire))**

On appelle un chromosome ou individu A est suite de bits en codage binaire de longueur l(A) où$A = \{a_1, a_2, \dots \dots, a_n\}$avec$i \in [1, n]eta_i \in \{0,1\}$.

- **Définition 2(Fitness d'un chromosome)**.

On appellefitness d'un chromosome toute valeur qu'on note f (A), où f est typiquement appelée fonction de fitness.

b) Les phases d'exécution des algorithmes génétiques

Les algorithmes génétiques sont alors basés sur les phases suivantes :

1. Initialisation. Une population initiale de N chromosomes est tirée aléatoirement.
2. Évaluation. Chaque chromosome est décodé, puis évalué.
3. Sélection. Création d'une nouvelle population de N chromosomes par l'utilisation d'une méthode desélection appropriée.
4. Reproduction. Possibilité de croisement et mutation au sein de la nouvelle population.
5. Retour à la phase d'évaluation jusqu'à l'arrêt de l'algorithme.

c) Eléments fondamentaux des algorithmes génétiques

- **Codage d'une population :**Le premier pas dans l'implémentation des algorithmes génétiques est de créer une population d'individus initiaux. En effet, les algorithmes génétiques agissent sur une populationd'individus, et non pas sur un individu isolé. Par analogie avec la biologie, chaque individu de la population est codé par un chromosome appelé aussi génotype. Unepopulation est donc un ensemble de chromosomes où chaque chromosome code un pointde l'espace de recherche. L'efficacité de l'algorithme génétique va donc dépendre duchoix du codage d'un chromosome. Il existe trois principaux type de codage : binaire, gray ou réel.

- **Fonction De Fitness :**pour calculer le coût d'un point de l'espace de recherche, on utilise une fonction d'évaluation appelée fonction fitness ou objectif. L'évaluation d'un individu ne dépend pas de celle des autres individus. Le résultat fourni par la fonction d'évaluation va permettre de sélectionner ou de refuser un individu pour ne garder que les individus ayant le meilleur coût en fonctionde la population courante. Cette méthode permetde s'assurer que les individus performants seront conservés, alors que les individus peu adaptés seront progressivement éliminés de la population.

- **Croisement (Hybridation) :**L'opérateur de croisement permet la création de nouveaux individus appelés fils à partir de deux individus dits parents. Les nouveaux individus héritent certaines caractéristiques de leurs parents. La probabilité d'hybridation représente la fréquence à laquelle les hybridations sont appliquées. S'il n'y a pas d'hybridation, les fils sont l'exacte copie des parents. S'il y a hybridation, les fils sont composés d'une partie de chacun de leurs parents. Si la probabilité est de 0%, la nouvelle génération est la copie de la précédente. Si la probabilité est fixée à 100%, tous les descendants sont générés par hybridation. L'hybridation est mise en place pour que les nouveaux chromosomes gardent la meilleure partie des chromosomes anciens. Ceci dans le but d'obtenir, peut-être, les

meilleurs chromosomes. Néanmoins, il est quand même important qu'une partie de la population survive à la nouvelle génération.

- **Mutation :**Le rôle de cet opérateur est de modifier aléatoirement, avec une certaine probabilité, la valeur d'un gène de l'individu. Dans le cas du codage binaire, chaque bit $a_i \in \{0, 1\}$ s'il est concerné par l'opération de mutation, sera remplacé selon une probabilité p_m par son inverse $a_i' = 1 - a_i$. Tout comme plusieurs lieux de croisement peuventêtre possibles, nous pouvons très bien admettre qu'une même chaîne puisse subir plusieurs mutations. Pour éviter à l'AG de converger vers des extrêmeslocaux de la fonction et de permettre de créer des éléments originaux. Si la mutation génèreun individu plus faible l'individu sera éliminé. La probabilité de mutation représente la fréquence à laquelle les gènes d'un chromosome sont mutés. S'il n'y a pas de mutation, le fils est inséré dans la nouvelle population sans changement. Si la mutation est appliquée,une partie du chromosome est changée.
- **Itération :** A partir d'une population initiale de n individus, l'algorithme génétique sélectionne une populationintermédiaire de **m** individus en faisant une sélection sur la population initiale (même individu peut être sélectionné plusieurs fois comme jamais il pourra être sélectionné, enfonction de la valeur de sa fonction d'évaluation). Les **m** individus de la population se croisent deux à deux (les couples se forment aléatoirement) pour construire n nouveauxindividus. Ces individus passent par un opérateur de mutation qui agit aléatoirement avec une possibilité faible 2-3% de bits pour former une nouvelle population. On réitère ensuite le procédé à partir de cette population jusqu'à obtenir une solution que l'on jugesatisfaisante.La sélection des individus qui interviennent dans l'opération de croisement prend plusieurs formes :
    - **Sélection par roulette (wheel) :**Les parents sont sélectionnés en fonction de leur performance. Par conséquent, les parents ayant de meilleures performances ont plus de chance d'être sélectionnés. Il fautimaginer une sorte de roulette de casino sur laquelle sont placés tous les chromosomesde la population, la place accordée à chacun des chromosomes étant en relation avecsa valeur d'adaptation. Ensuite, la bille est lancée et s'arrêtée sur un chromosome. Lesmeilleurs chromosomes peuvent ainsi être tirés plusieurs fois et les plus mauvais aurontune très petite chance d'être sélectionnés. Cela peut être simulé par l'algorithme suivant: On calcule la somme S_1 de toutes les fonctions d'évaluation d'une population. Ongénère un nombre **R** entre 0 et S_1. On calcule ensuite une somme S_2 des évaluations ens'arrêtant dès que **R** est dépassé. Le dernier chromosome dont la fonction d'évaluationvient d'être ajouté et sélectionné.
    - **Sélection par rang :**La sélection précédente rencontre des problèmes lorsque la valeur d'adaptation des chromosomes varieénormément. Si la meilleure fonction d'évaluation d'un chromosomereprésente 90% de la roulette alors les autres chromosomes auront très peu de chanced'être sélectionnés et on arriverait à une stagnation de l'évolution. Cette sélection trie d'abord la population par fitness. Ensuite, chaque chromosome se voit associé unrang en fonction de sa position. Ainsi le plus mauvais chromosome aura

le rang 1, lesuivant 2, et ainsi de suite jusqu'au meilleur chromosome qui aura le rang N (N représente le nombre d'individus dans la population). La sélection par rang d'un chromosome est la même que par roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation.

- **Selection steady-state :** L'idée principalede cette méthode de sélection est qu'une grande partie de la population puisse survivre à la prochaine génération.L'algorithme génétique fonctionne alors de la manière suivante : à chaque génération sont sélectionnés quelques chromosomes (parmi ceux qui ont le meilleur coût) pour créer deschromosomes fils. Ensuite les chromosomes les plus mauvais sont retirés et remplacéspar les nouveaux. Le reste de la population survie à la nouvelle génération.
- **Sélection par tournoi :** Cette méthode n'utilise que des comparaisons entre individus et ne nécessite pas le tri de la population. Elle possède un paramètre, taille du tournoi. Pour sélectionner un individu, on en tire t uniformément dans la population, et on sélectionne le meilleur de ces t individus. S t=2 alors on effectue un tirage avec remise de deux individus de la population, et on les fait "combattre". Celui qui a l'adaptation la plus élevée l'emporte avec une probabilité p comprise entre 0.5 et 1. On répète ce processus n fois de manière à obtenir les n individus de la population qui serviront de parents. Le choix de t permet de faire varier la pression sélective, c'est-à-dire les chances de sélection des plus performants par rapport aux plus faibles.
- **Elitisme :**A la création d'une nouvelle population, il y a de grandes chances que les meilleurs chromosomessoient perdus après les opérations d'hybridation et de mutation. Pour évitercela, on utilise la méthode d'élitisme. Elle consiste à copier un ou plusieurs des meilleurschromosomes dans la nouvelle génération. Ensuite, on génère le reste de la populationselon l'algorithme de reproduction usuel. Cette méthode améliore considérablement lesalgorithmes génétiques, car elle permet de ne pas perdre les meilleures solutions.

d) Pseudo code d'algorithme génétique

Les algorithmes génétiques s'exécutent selon le pseudo code suivant :

Début

- Population initial composé de m solution : $P_0 = \{x_0, x_1, \ldots\ldots, x_m\}$
- K = 0

Tant Que (condition d'arrêt non vérifiée)

- Appliquée un opérateur de sélection pour obtenir des paires de solutions de la population P_K. '' (y_1, z_1), …… , (y_p, z_p) ''
- $P_{K+1} = \emptyset$

Pour j de 1 à p

- Appliqué l'opérateur de croisement pour (y_j, z_j) pour obtenir $X(y_j, z_j)$ l'ensemble des nouvelles solutions
- Pour chaque solution $x \in X(y_j, z_j)$ appliqué l'opérateur de mutation pour obtenir $\mu(x)$
- $P_{K+1} = P_{K+1} \cup \mu(x)$

Fin pour

- K=k+1

Fin Tant Que

Fin

2.4 Conclusion

Dans ce chapitre, nous avons présenté l'optimisation combinatoire et la classification des méthodes de résolution pour les problèmes d'optimisation combinatoire. Nous avons mis l'accent sur deux métaheuristiques que nous allons utiliser dans notre projet de fin d'études : les algorithmes génétiques et le recuit simulé.

Le chapitre suivant détaille l'application de ces deux métaheuristiques pour minimiser le nombre des RSUs tout en gardant des critères de performances acceptables pour la dissémination des informations dans les VANETs

Chapitre III

Déploiement efficace des RSUs dans lesréseaux VANETs

Chapitre 3 Déploiement efficace des RSUs dans les réseaux VANETs

3.1 Introduction

Pour garantir une dissémination efficace des informations dans les réseaux VANETs en particulier dans un environnement urbain, il est recommandé de placer un nombre suffisant de RSUs au niveau des intersections. De ce fait, nous avons proposé dans le cadre de ce projet de fin d'études de trouver un nombre minimal de RSUs tout en garantissant une performance de routage acceptable. Et vu que le nombre de combinaisons est très grand alors que les méthodes exactes ne peuvent pas résoudre ce problème considéré comme un problème d'optimisation combinatoire d'une manière efficace. D'où nous avons fait recours à des métaheuristiques qui ont prouvé leur efficacité pour traiter ce type de problèmes. Une des métaheuristiques fait partie des méthodes à solution unique (recuit simulé) et une autre méthode à population de solutions (les algorithmes génétiques). Ces deux métaheuristiques sont réalisées en deux versions chacune. La première version est basique et la deuxième est une version améliorée pour accélérer la convergence.

La résolution de ce problème se fait en deux phases : la première phase est réalisé en Java sous l'IDE éclipse pour trouver le nombre de RSUs et la deuxième phase consiste à injecter ces résultats dans une autre application développée en OMNET++ pour évaluer les performances de routage.

Dans ce chapitre, nous exposons le problème abordé dans le cadre de ce PFE. Puis, nous présentons notre démarche de résolution et les outils nécessaires pour ce faire.

3.2 Problématique

Dans le cadre de notre projet de fin d'études qui est une amélioration du PFE [1], nous avons visé à minimiser le nombre des RSUs tout en gardant une performance de routage acceptable. Ces RSUs sont placés au milieu des intersections. Nous avons pris comme exemple d'application la localité d'Abou Tachfine de la wilaya de Tlemcen comme nous pouvons l'appliquer à n'importe quelle autre localité.

Nous avons modélisé notre démarche comme montre la figure III.1.

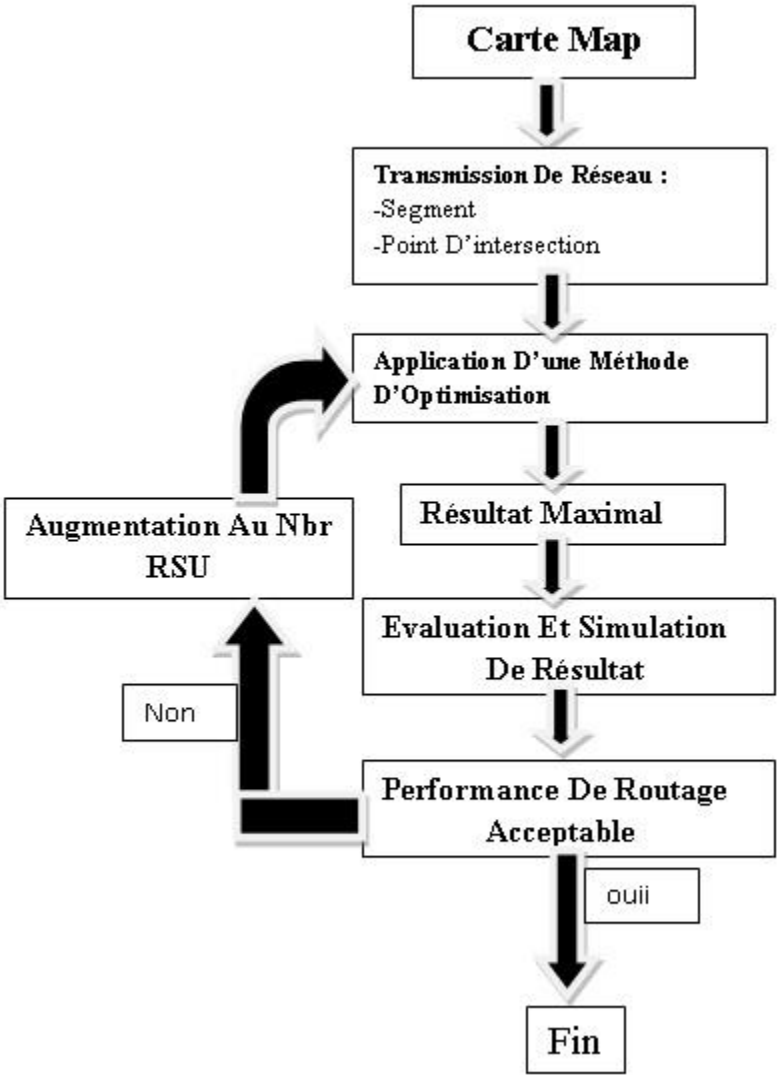


Figure III.1: Modélisation de la problématique

3.3 Solution proposée

Pour traiter cette problématique, nous proposons deux métaheuristiques une de chaque famille :

- Les algorithmes génétiques,
- Recuit simulé

3.3.1 Modélisation

Avant d'entamer les deux méthodes de résolution, nous modélisons ce problème. Dans ce cas, la modélisation consiste à représenterla zone urbaine concernée par des segments et des points d'intersection, où un segment représente une route entre deux intersections. Par ailleurs, la taille du chromosome est égal au nombre d'intersections au niveau de la zone urbaine. Le placement d'une RSU au niveau d'une intersection se fait comme suit :

- bit**1 : Représente qu'au niveau de cette intersection une RSU est placée.**
- bit **0 : Représente qu'il n'y a pas une RSU au niveau d'une intersection.**

3.3.2 Fonction fitness

La fonction fitness consiste à calculer la couverture de l'ensemble des RSUs de la zone urbaine prise en considération. Cette évaluation se fait d'une manière mathématique pour chaque intersection qui porte une RSU. La Figure III.2 représente le fonctionnement de cette fonction.

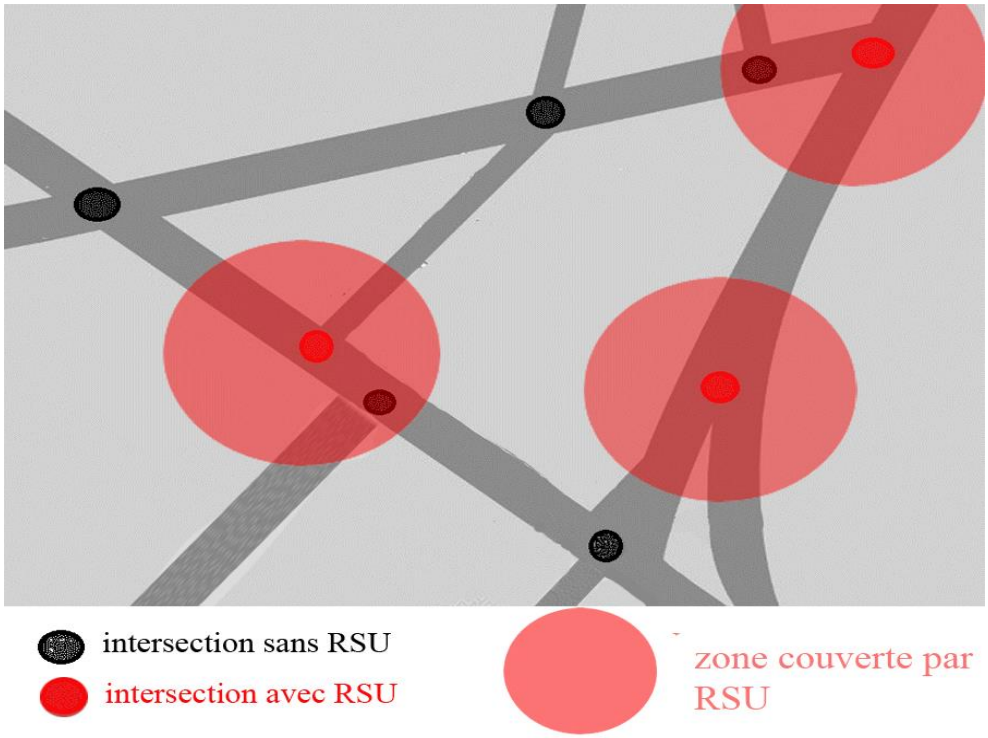


Figure III.2: Couverture par les RSUs

Le calcul de couverture au voisinage se fait selon le code de la Figure III.3.

```
public ArrayList<Segment>  CouvreVois(ArrayList<Segment> seg){
    int seg1=-1;
    for(int i=0;i<tabV.size();i++) {
        double x=this.getP().getX(),y=this.getP().getY();
        if(this.getP().getX()<tabV.get(i).getP().getX()){
            while(x<=tabV.get(i).getP().getX() && Dist(new Poit(x,y), this.getP())<=FenetrAction.capaciter){
                x+=0.1;
                y=ValMost(this.getP(), tabV.get(i).getP(), x);
            }
            seg1=this.CherchSeg(seg, i);
            if(seg1!=-1)
                seg.get(seg1).SegCouvr(this, this.getP(), new Poit(x,y));
        }
        if(this.getP().getX()>tabV.get(i).getP().getX()){
            while(x>=tabV.get(i).getP().getX() && Dist(new Poit(x,y), this.getP())<=FenetrAction.capaciter){
                x-=0.1;
                y=ValMost(this.getP(), tabV.get(i).getP(), x);
            }
            seg1=this.CherchSeg(seg, i);
            if(seg1!=-1)
                seg.get(seg1).SegCouvr(this, this.getP(), new Poit(x,y));
        }
        if(this.getP().getX()==tabV.get(i).getP().getX()){
            if(this.getP().getY()<tabV.get(i).getP().getY()){
                while(y<=tabV.get(i).getP().getY() && Dist(new Poit(x,y), this.getP())<=FenetrAction.capaciter){
                    y+=0.1;
                }
                seg1=this.CherchSeg(seg, i);
                if(seg1!=-1)
                        seg.get(seg1).SegCouvr(this, this.getP(), new Poit(x,y));
                }
                if(this.getP().getY()>tabV.get(i).getP().getY()){
                    while(y>=tabV.get(i).getP().getY() && Dist(new Poit(x,y), this.getP())<=FenetrAction.capaciter){
                        y-=0.1;
                    }
                    seg1=this.CherchSeg(seg, i);
                    if(seg1!=-1)
                        seg.get(seg1).SegCouvr(this, this.getP(), new Poit(x,y));
                }
            }
        }

        return seg;
    }
```

Figure III.3: Fonction pour couverture de Voisinage

Le calcul de la couverture d'un segment (route) se fait comme suit (Figure III.4) :

```
public ArrayList<Segment> Couverture(ArrayList<Segment> seg){
    double Tx[],Ty[]=new double[2];
    boolean tst=false;
    for(int i=0;i<seg.size();i++){
        tst=SegRSU(seg.get(i));
        if(tst==false){///// les taravail de couverture
            if(seg.get(i).p2.getX()-seg.get(i).p1.getX()!=0){
            Tx=CalculeDuCercle(seg.get(i), this.getP(), FenetrAction.capaciter*FenetrAction.capaciter);
                if(Tx!=null){
                    Ty[0]=ValMost(seg.get(i).p1,seg.get(i).p2 , Tx[0]);
                    Ty[1]=ValMost(seg.get(i).p1,seg.get(i).p2 , Tx[1]);

                    int index1=TrouveInedex(seg.get(i), Tx[0]),index2=TrouveInedex(seg.get(i), Tx[1]);
                    if(index1<index2){
                        for(int j=index1;j<=index2;j++)
                            seg.get(i).Ss.get(j).RsuAjouter(this);

                    }else {
                        for(int j=index2;j<=index1;j++)
                            seg.get(i).Ss.get(j).RsuAjouter(this);

                    }
                }
            }else {
                Ty=CalculeDuCercle(seg.get(i), this.getP(), FenetrAction.capaciter*FenetrAction.capaciter);
                if(Ty!=null){

                    int index1=TrouveInedexY(seg.get(i), Ty[0]),index2=TrouveInedexY(seg.get(i), Ty[1]);
                    if(index1<index2){
                        for(int j=index1;j<=index2;j++)
                            seg.get(i).Ss.get(j).RsuAjouter(this);

                    }else {
                        for(int j=index2;j<=index1;j++)
                            seg.get(i).Ss.get(j).RsuAjouter(this);

                    }
                }
            }
        }

    }
    return seg;
}
```

Figure III.4: Fonction pour couverture d'une route

- **Couverture de Voisinage :** L'algorithme prend la table de voisinage de l'intersection qui contient une RSU et il calcule la somme des distances qui sont inférieures ou égales à la portée d'une RSU pour chaque segment de la table de voisinage.
- **Couverture d'un segment (route) :**L'algorithme prend le point d'intersection qui contient une RSU comme un centre d'un cercle dont le rayonest égal à la portée de transmission d'une RSU. Puis il cherche parmi les segments du réseau les segments qui appartiennent au cercle (il calcule la somme de toutes les distances trouvées par l'algorithme).
- **Couverture totale :** La couverture totale représente la somme de toutes les distances trouvéesà l'aide des deux algorithmes de couverture.

Evaluation de chevauchement entre RSUs

Après le calcul de la couverture pour chaque intersection qui contient une RSU, on aura des segments qui sont couverts par plus d'une RSU i.e. les segments là où il y a des chevauchements. Dans ce cas, on applique une méthode simple qui permet de diviser un segment en plusieurs sous-segments comme montre la figure III.6 où chaque sous-segment est de longueur d'un mètre. Le calcul de chevauchement se fait par le code représenté par la figure III.7.

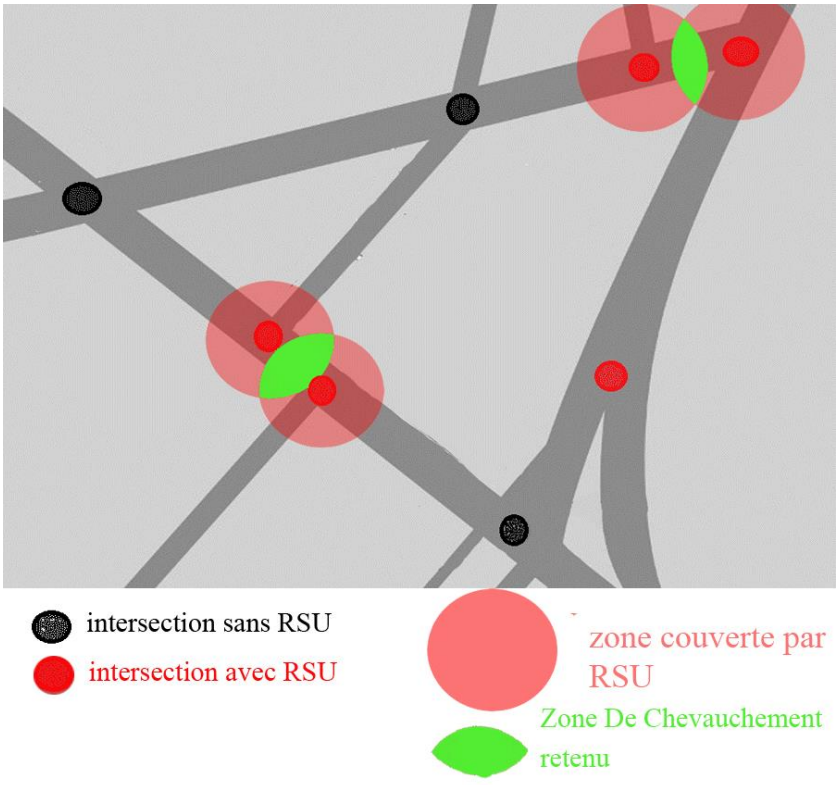
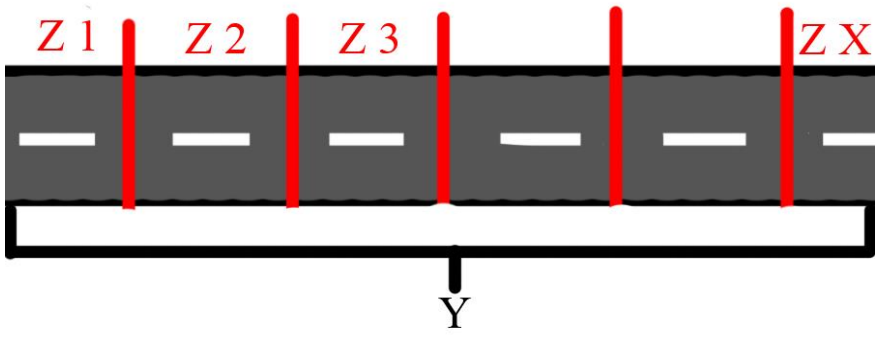


Figure 8: Chevauchement entre RSUs



Y : segment

Z1 Z2 Z3.....Zx :sous segements

Figure III.6: Division d'un segment en sous-segments

```
public int TotalChauvochement(ArrayList<Segment> seg){
    int conteur=0;
    int j;
    boolean tst=false;
    for(int i=0;i<seg.size();i++){
        j=0;
        while(j<seg.get(i).Ss.size()){
            if(seg.get(i).Ss.get(j).getCpt()>1){
                tst=true;
                j=seg.get(i).Ss.size();
            }
            j++;
        }
        if(tst==true){
            conteur++;
            tst=false;
        }
    }
    return conteur;
}
```

Figure III.7: Calcul de chevauchement

- **Délai**

A la fin on aura un problème de délai de transmission, car on peut tomber sur deux intersections qui ont une même couverture mais le délai de transmission n'ai pas le même alors on applique cette fonction pour départager et améliorer par la suite la solution trouvée.

3.3 Résolution par les algorithmes Génétiques

L'exécution de cette métaheuristique se fait de deux manières : une manière basique et une autre améliorée dans le but d'accélérer la convergence

3.3.1 Version basique des AGs

Dans ce cas on applique la version originale de l'algorithme génétique. D'où on aura une initialisation de la population d'une manière aléatoire, puis on entame l'évaluation, en suite la sélection, après on entame le croisement et mutation jusqu'à ce qu'on obtienne une meilleure solution parmi les solutions explorées.

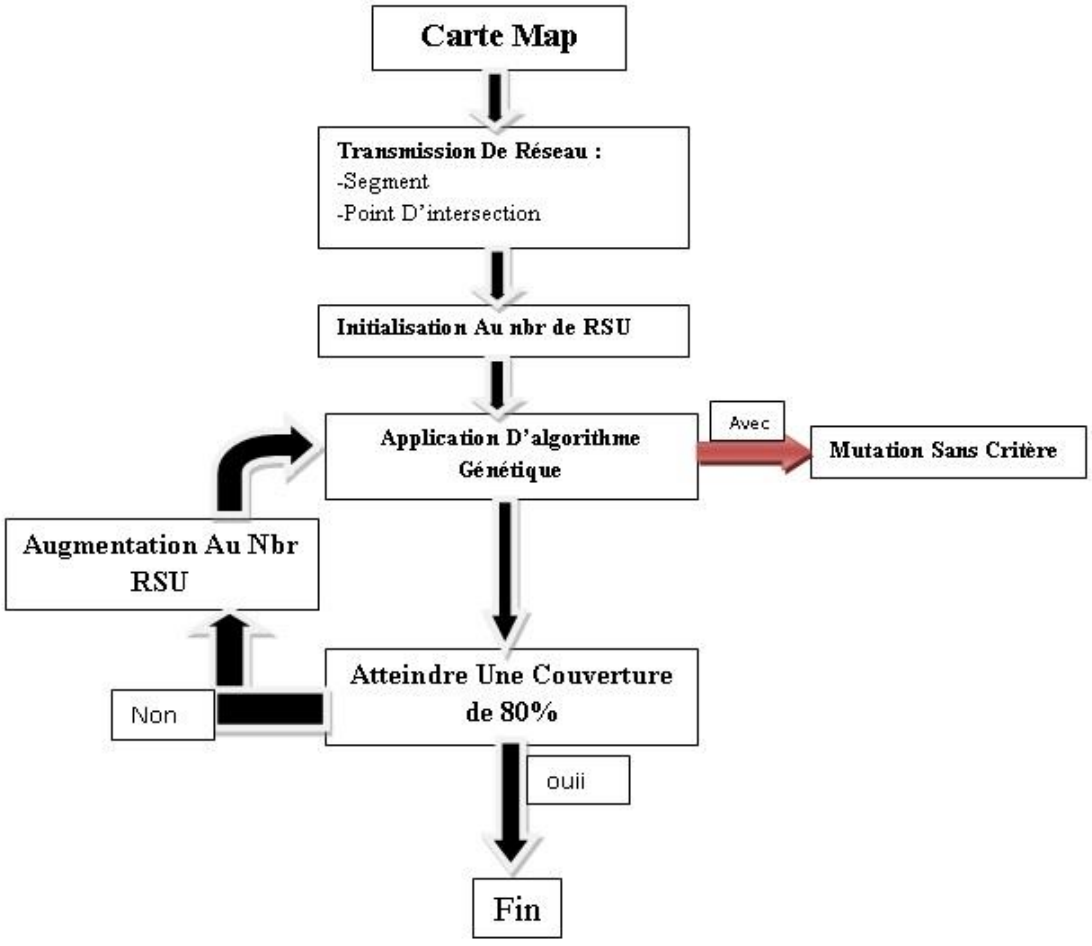


Figure III.8: La version basique des algorithmes génétiques

```
public void run(){
    int nbrIntr=net.getRSU().size();
    PopulationInicial p;
    Evaluation ev;
    ArrayList<ResultatAlgo> rAl =rAl=new ArrayList<>();
    double taut=0;
    while(taut<80){
        FenetrAction.nbrRSUs=FenetrAction.nbrRSUs+nbrRSUs;
    p=new PopulationInicial(net,FenetrAction.nbrPop , FenetrAction.nbrRSUs, nbrIntr);
    ev=new Evaluation(p.Pop, net);
    int i=0;
    while(i<ev.Pop.size()){
        rAl.add(new ResultatAlgo(ev.Pop.get(i),p.RsltCouverture(ev.Pop.get(i), net) ,
                p.RsltChauvochement(ev.Pop.get(i), net),ev.resultat.get(i)));
        i++;
    }

    Selection s=new Selection(rAl);
    rAl=s.rAl;
    int cpt=0;
    int n=0;
    while(n<10000){
        double r1=0,r3,r4;
        Croisement cc=new Croisement(s.Per,s.Mer,net);
        if(FenetrAction.basic!=true){
            r1=ev.FonctionObjct(s.Per);
            r3=ev.FonctionObjct(cc.Fils1);
            r4=ev.FonctionObjct(cc.Fils2);
        }else{
            r1=ev.FonctionObjct2(s.Per);
            r3=ev.FonctionObjct2(cc.Fils1);
            r4=ev.FonctionObjct2(cc.Fils2);
        }
        if(r3>=r4){
             if(r3>r1){
                 cpt++;
                 rAl.set(0, new ResultatAlgo(cc.Fils1
                         ,p.RsltCouverture(cc.Fils1, net),p.RsltChauvochement(cc.Fils1, net),r3));
             }
         }
         if(r3<=r4){
             if(r4>r1){
                 cpt++;
                 rAl.set(0, new ResultatAlgo(cc.Fils2
                         ,p.RsltCouverture(cc.Fils2, net),p.RsltChauvochement(cc.Fils2, net),r4));
             }
         }
         if(cpt==30)
             n=10000;
         s=new Selection(rAl);
         n++;
     }
    }
```

Figure III.9: Code de la version basique des AGs

3.3.2 Version améliorée des AGs

Dans la version améliorée, un traitement préalable est réalisé sur une telle génération là où la population contient les individus dont le taux de chevauchement entre eux est très faible. La figure illustre la version améliorée des AGs.

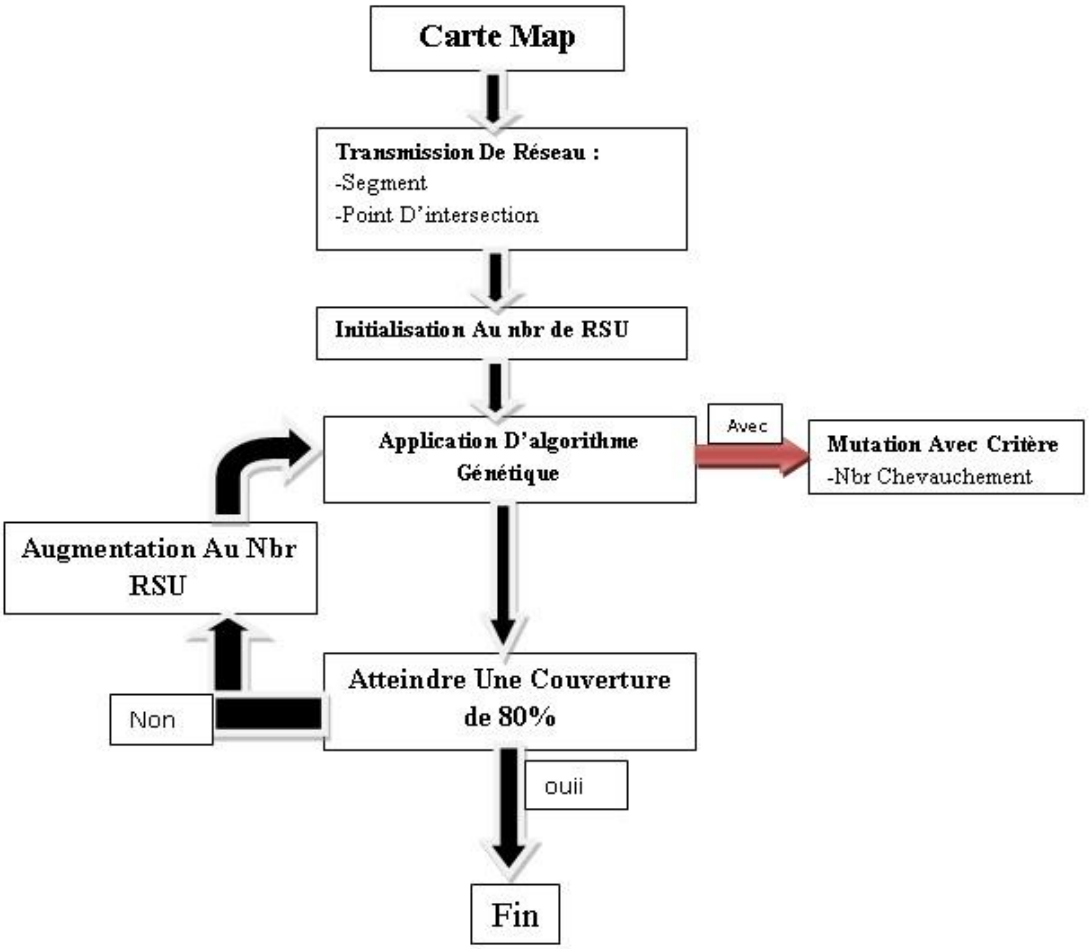


Figure III.10: La version améliorée des AGs

```
public Mutation(int f1[] ,int f2[], int cpt,Network net){

    int y=(int) Math.floor(Math.random()*f1.length);
    if(f1[y]!=1)
        f1[y]=1;
    else f1[y]=0;

    if(f2[y]!=1)
        f2[y]=1;
    else f2[y]=0;
if(FenetrAction.basic!=true){
    while(TrouvChauv(f1, net)<TrouvChauv(Fils1, net) || TrouvChauv(f2, net)<TrouvChauv(Fils2, net)){
        y=(int) Math.floor(Math.random()*f1.length);
        if(f1[y]!=1)
            f1[y]=1;
        else f1[y]=0;
        if(f2[y]!=1)
            f2[y]=1;
        else f2[y]=0;

        }
      }
    }
```

Figure III.11: Partie code de la version améliorée des AGs

3.4 Résolution par la métaheuristique Recuit Simulée

Cette méthode, contient un paramètre de contrôle (Température). Ce paramètre est abstrait et il doit être initialisé suivant l'équation ci-dessous :

$$T_0 = 1.44 * M$$

Où M représente la valeur médiane. Cette valeur de M est calculée après avoir générer aléatoirement un certain nombre de configurations puis calculer le coût de chacune et faire un tri en fonction de ces coûts. Enfin, on prend la valeur médiane (la valeur qui occupe le milieu).

La dégradation de la température lors du passage d'un certain nombre d'itérations à un autre se fait comme suit :

$$T \longleftarrow T * 0.81$$

Ce paramètre doit être décrémenté jusqu'à ce qu'il soit égal à epsilon (epsilon = 0.00001).

3.4.1 Version basique du recuit simulé

Dans cette version, nous choisissons le voisin le plus proche à la configuration actuelle sans tenir compte du critère de chevauchement selon l'organigramme représenté par la figure 21.

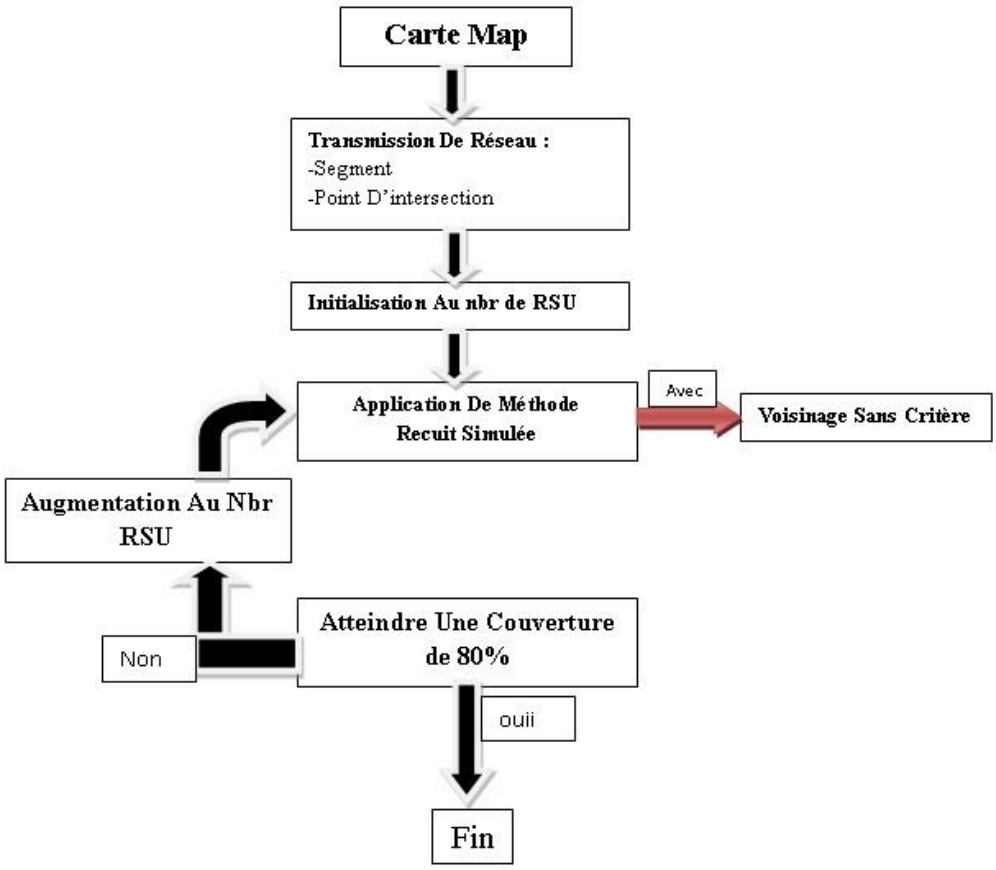


Figure III.12: Version basique du recuit simulé

```
public void run(){
    AlgoritmeRecuitSimuler alg1 = null;
    ResultatAlgo rAl = null;
    int tab1[] = null;
    double epsilon = 0.00001;
    double mu = 0.87;
    double taut=0;
    int cpt=0;
    in=0;
    while(taut<80){

        FenetrAction.RsltTexte.setText(cpt+"Patienter SVP .");
        in=in+FenetrAction.nbrRSUsInitial;
        alg1=new AlgoritmeRecuitSimuler(net,in);
        c= 1.44*net.Mediom();
    tab1=alg1.Initialisation(alg1.getChromosome(),in);
     if(FenetrAction.basic!=true){
            rAl=new ResultatAlgo(tab1, alg1.FonctionObjectiv(tab1));
        }else rAl=new ResultatAlgo(tab1, alg1.FonctionObjectiv2(tab1));
    resultMax=alg1.FonctionObjectiv(tab1);
    tabMax=tab1;

        while(c>epsilon){
            FenetrAction.RsltTexte.setText(cpt+"Patienter SVP ...");
            int i=0;
            while(i<100/*net.getRSU().size()*/){
                FenetrAction.RsltTexte.setText(cpt+"Patienter SVP ......");
                int tab2[]=Vosinage(tab1);
                ResultatAlgo rAl1;
                if(FenetrAction.basic!=true){
                    rAl1=new ResultatAlgo(tab2, alg1.FonctionObjectiv(tab2));
                }else rAl1=new ResultatAlgo(tab2, alg1.FonctionObjectiv2(tab2));
                if(rAl.rslt<rAl1.rslt){
                    FenetrAction.RsltTexte.setText(cpt+"Patienter SVP ........");
                    rAl=rAl1;
                      resultMax=alg1.FonctionObjectiv2(tab2);
                      tabMax=tab2;
                  }else if(rAl1.rslt<rAl.rslt){
                      FenetrAction.RsltTexte.setText(cpt+"Patienter SVP ...........");
                      double R=Math.random();
                      double deltat=(rAl1.rslt-rAl.rslt)/c;
                      if(R<=Math.exp(deltat)){
                          rAl=rAl1;
                      }
                  }

                  i++;
              }
              c=c*mu;
          }
          cpt++;
      Resultat=alg1.FonctionObjectiv2(rAl.tab);
      taut =(100*Resultat)/net.DistanceTotale();
  }
```

Figure III.13: Partie code de la version basique du recuit simulé

3.4.2 Version améliorée du recuit simulé

Dans version lors de l'exécution de la fonction voisinage V(X) on s'intéresse particulièrement aux voisins qui ne causent pas un grand chevauchement. L'exécution de cette version se fait selon l'organigramme ci-dessous.

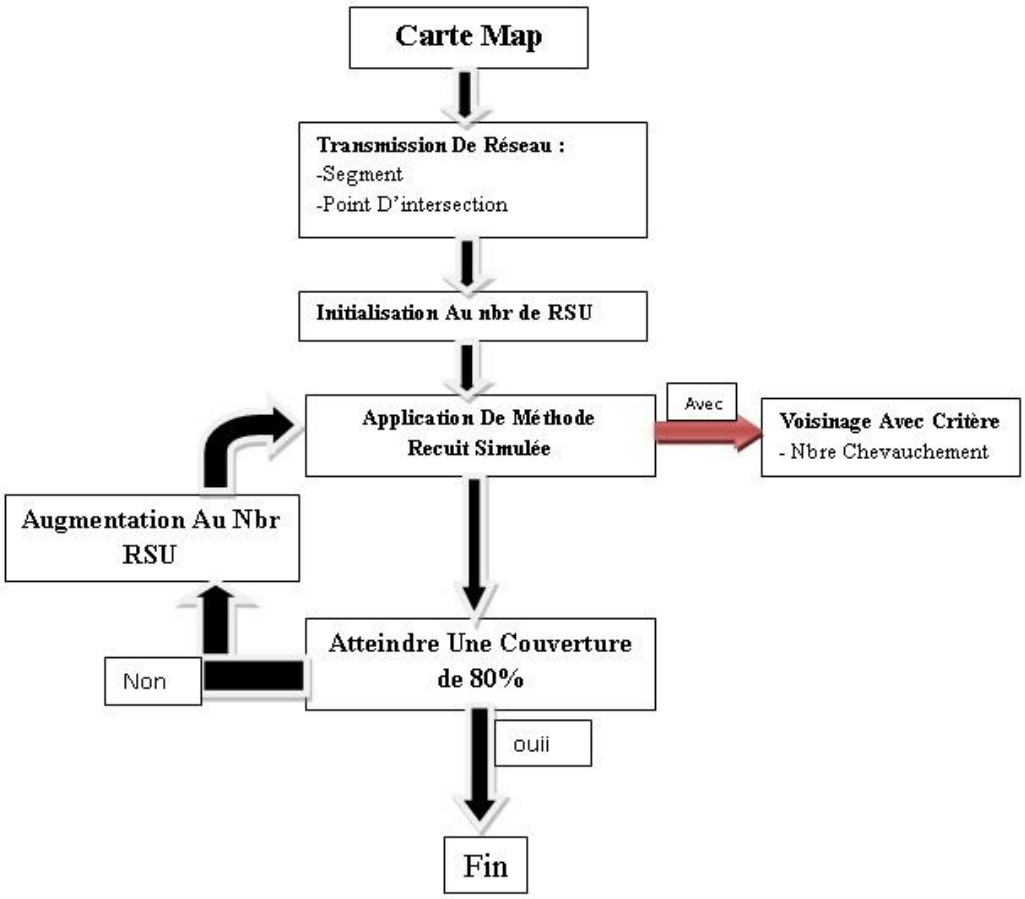


Figure III.14: Organigramme de la version améliorée du recuit simulé

```
        while(TrouvChauv(tab)<TrouvChauv(tab1)){
            x=(int) (Math.round(Math.random()*(tab.length-2)+1));
            y=(int) (Math.round(Math.random()*(tab.length-2)+1));
            while(x==y){
                y=(int) (Math.round(Math.random()*(tab.length-2)+1));
            }
            if(tab1[x]==0){
                tab1[x]=1;
                t1=true;
            }else if(tab1[x]==1){
                tab1[x]=0;
                t0=true;
            }
            if(tab1[y]==0){
                if(t1!=true)
                    tab1[y]=1;
            }else if(tab1[y]==1){
                    if(t0!=true)
                        tab1[y]=0;
                }
            tab1=ReglageUn(tab, in);
        }
    }
    tab=tab1;
    return tab;
}
```

Figure III.15: Partie du code de la version améliorée du recuit simulé

3.5 Evaluation et Simulation

3.5.1 Calcul du nombre de RSUs

Tout d'abord, on récupère la map de la localité d'Abou Tachfine comme montre la figure ci-dessous.

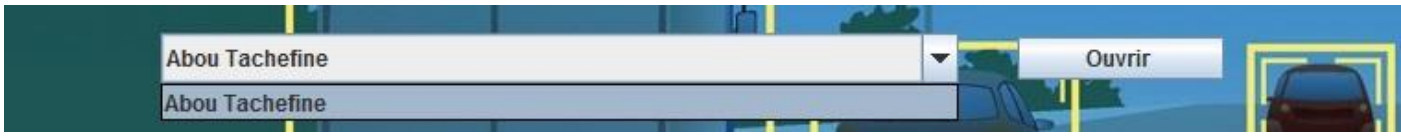


Figure III.16: Récupération du map de la localité d'Abou Tachfine

Puis on choisit une méthode à utiliser selon l'interface ci-dessous :

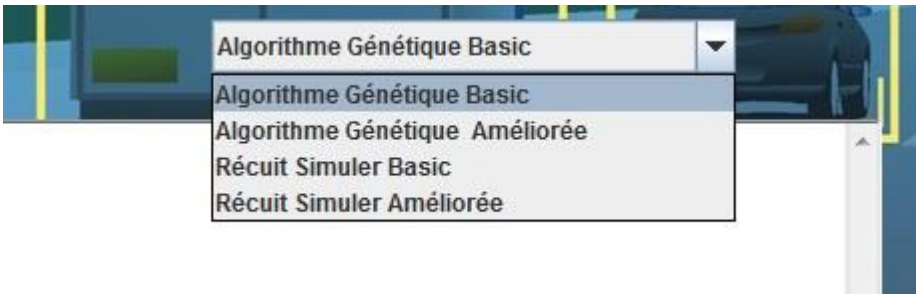


Figure III.17: Choix de la méthode de résolution

Figure III.18: Interface de la méthode AG

Figure III.19: Interface de la méthode Recuit Simulé

Apres un certain moment on aura un affichage des résultats sur l'interface

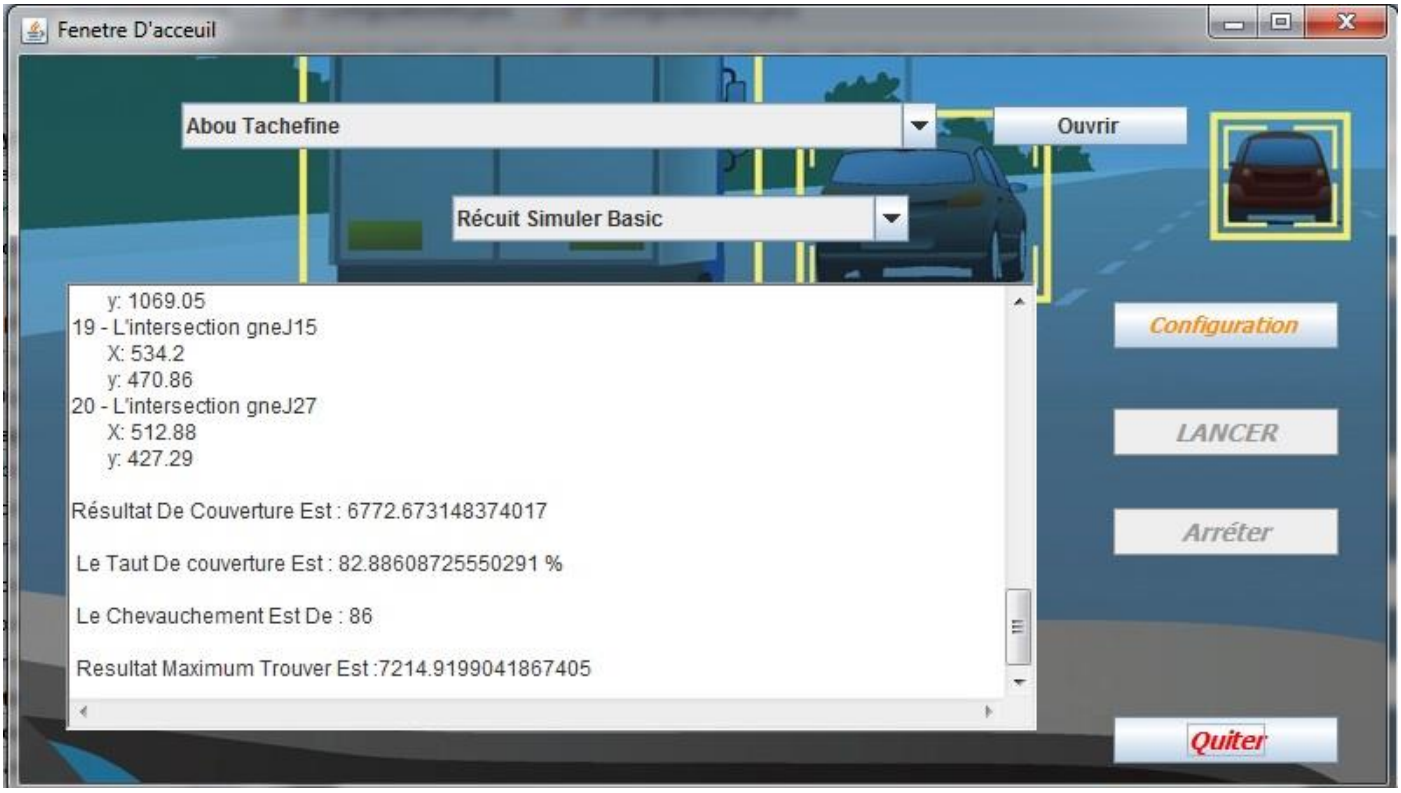


Figure III.20: Résultats de Simulation après exécution

Après cette résultat il faut qu'on passe à la simulation, pour simulé la solution et savoir si la performance de routage est acceptable ou non. Si oui on arrête si non on augmente le nombre de RSU pour avoir une autre solution qui garde cette performance.

3.5.2 Evaluation des performances de routage

Dans notre projet de fin d'étude, l'objectif principal est de trouver un nombre minimal des RSUs à déployer et leurs positions optimales tout en respectant certaines exigences de routage. Pour ce faire, nous proposons les étapes de traitement comme il est représenté sur la Fig. III.21.

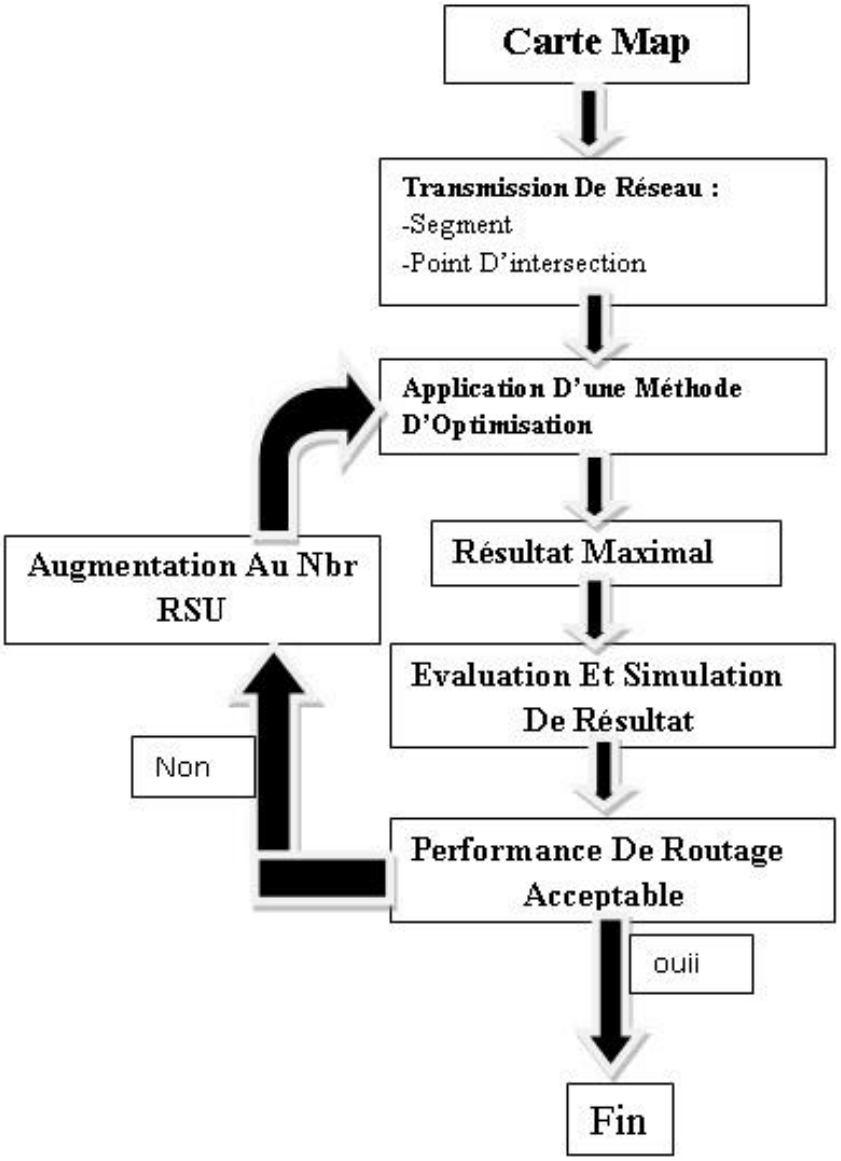


Figure III.21: Diagramme d'évaluation de la qualité de routage

Pour ce faire, on a utilisé les deux méthodes d'optimisation précédentes, et pour évaluer les performances de nos solutions, nous choisissons deuxcritères :

1) le taux de livraison pour envoyer des messages de n'importe quel segment routier vers la RSU la plus proche,
2) la zone de couverture minimalepour garantir letraitement des demandes des véhicules.

Dans notre évaluation, nous avons choisi le protocole de routage développé dans [11]. Les tableaux 1 et 2 illustrent les résultats obtenus par chaque métaheuristique.

Tableau 1: Résultats de simulation en utilisant les AGs

Nombre RSU	Zone de couverture (m^2)	Chevauchement ($m^{2)}$	Taux de couverture	Taux de traitement de demande des véhicules
10	6284	58	76.90%	80.63%
20	7564.12	90	92.57%	94.74%
30	8073.19	111	98.80%	100.00%
40	8131.31	130	99.51%	100.00%
50	8171.061	131	100.00%	100.00%
60	8171.06	133	100.00%	100.00%
70	8171.06	133	100.00%	100.00%
80	8171.06	134	100.0 %	100.00%

Nombre RSU	Zone de couverture (m^2)	Chevauchement ($m^{2)}$	Taux de couverture	Taux de traitement de demande des véhicules
10	5777.90	48	70.71%	74.32%
20	7220.36	81	88.36%	90.54%
30	7765.80	113	95.04%	97.29%
40	8063.02	115	98.67%	99.78%
50	8158.89	129	99.85%	100.00%
60	8171.06	127	100.00%	100.00%
70	8171.06	132	100.00%	100.00%
80	8171.06	134	100.00	100.00%

Tableau 2: Résultats de simulation en utilisant l'algorithme Recuit simulé

Dans le tableau.1, on remarque quand le nombre est 30 RSUs, le taux de couverture atteint 98.80% et le taux de traitement des demandes des véhicules atteint la 100%. D'où on remarque que la performance de routage est gardé à 100% quand le nombre des RSUs est réduit à 30.

Dans le tableau.2, on remarque qu'après une minimisation de 40% le nombre des RSUs, le taux de couverture atteint la 99.85% et le taux de traitement des demandes des véhicules atteint 100%. D'où, on remarque que la performance de routage est gardée à 100% après une minimisation le nombre de 40% le nombre des RSUs.

3.6 Conclusion

Dans ce chapitre, nous avons traité un problème d'optimisation combinatoire qui consiste à déployer un nombre minimal de RSUs dans une zone urbaine tout en gardant les performances

de routage acceptables. Pour ce faire, nous avons utilisons deux métaheuristiques : les algorithmes génétiques et le recuit simulé. Nous avons évalué le taux de couverture par un certain nombre de RSUs d'une manière incrémentale jusqu'à ce que ce taux soit au voisinage de 90%. Puis nous avons injecté ces résultats sous OMNET++ pour vérifier si les performances de routage sont acceptables en évaluant le taux de paquets "Request" sur les paquets "Replay". Les résultats obtenus étaient satisfaisants.

Conclusion générale

Conclusion Générale

Les réseaux VANETs sont un type particulier de réseaux sans fil dans lequel lesnœuds sont des véhicules en mouvement sur les routes. Ce type de réseaux est uneversion spécifique des réseaux mobiles ad-hoc MANET avec des spécificitéssupplémentaires. Ils déploient la communication et l'échange d'informations entre lesusagers de la route.

Pour assurer la dissémination des informations d'une manière efficace dans ce type de réseaux il est judicieux de mettre en place des RSUs au niveau des intersections. Or, le coût de ces RSUs est plus ou moins élevé ce qui nous oblige de minimiser leur nombre tout en garantissant des critères de routage acceptables.

Pour ce faire nous avons développé une application qui permet de déployer un nombre minimal de RSUs en utilisant deux métaheuristiques : les algorithmes génétiques et le recuit simulé. Les résultats obtenus sont prometteurs vu qu'on est arrivé à déployer moins de 50% des RSUs au niveau des intersections.

Bibliographie

Bibliographie

[1] Amrani Hanane et Sebaa Amel, “Conception et évaluation d’un protocole de routage basé sur EGYTAR pour les réseaux Vanets”, Mémoire de Master, Université de Tlemcen, 2017.

[2] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," in *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162-1182, July 2011.

[3] Y. Lin, P. Wang and M. Ma, "Intelligent Transportation System(ITS): Concept, Challenge and Opportunity," *2017 IEEE 3rd international conference on big data security on cloud (bigdatasecurity), IEEE international conference on high performance and smart computing (hpsc), and IEEE international conference on intelligent data and security (IDS)*, Beijing, 2017, pp. 167-172.

[4] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments," *VTC Spring 2008 - IEEE Vehicular Technology Conference*, Singapore, 2008, pp. 2036-2040.

[5] Hannes Hartenstein and Kenneth Laberteaux. VANET: Vehicular Applications and inter-networking technologies, volume 1. Wiley Online Library, 2010.

[6] Ali Hassoune Mustafa, “Multimédia QoS dans les réseaux sans fil : Etude de cas”, Thèse de Doctorat, Université des Sciences et de la Technologie d’Oran, 2018.

[7] IEEE Intelligent Transportation Systems Committee et al. IEEE trial-use standard for wireless access in vehicular environments (WAVE)-resource manager. IEEE Std 1609.1-2006, pages 1-71, 2006.

[8] Palpant M., "Recherche exacte et approchée en optimisation combinatoire : schémasd’intégration et applications", Thèse de Doctorat, Université d’Avignon, 2005.

[9] Reeves, C.: Modern Heuristic Techniques for Combinatorial Problems.Advances topics in computer science. Mc Graw-Hill, 1995.

[10] Omessaad, H : Contribution au développement de méthodes d’optimisation Stochastiques application à la conception des dispositifs électrotechniques, Thèse deDoctorat, Université de Lille France 2003.

[11] Nebbou T., Fouchal H., Lehsaini M., Ayaida M. (2017) A Realistic Location Service for VANETs. In: Eichler G., Erfurth C., Fahrnberger G. (eds) Innovations for Community Services. I4CS 2017. Communications in Computer and Information Science, vol 717. Springer, Cham, 2017.

Résumé

Pour assurer la dissémination des informations d'une manière efficace dans les réseaux VANETs il est recommandé d'opter pour une architecture dont les communications sont de type V2I où les RSUs seront au niveau des intersections. Or, le coût de ces RSUs est plus ou moins élevé ce qui nous oblige à minimiser leur nombre tout en garantissant des critères de routage acceptables.

Pour ce faire nous avons développé une application qui permet de déployer un nombre minimal de RSUs en utilisant deux métaheuristiques : les algorithmes génétiques et le recuit simulé. Les résultats obtenus sont prometteurs vu qu'on est arrivé à déployer moins de 50% des RSUs au niveau des intersections.

Abstract

To ensure the dissemination of information with an efficient manner in VANETs networks it is recommended to opt for an architecture whose communications are of type V2I wherein the RSUs will be placed at intersections. However, the cost of these RSUs is higher, which requires us to minimize their number while ensuring acceptable routing criteria.

For this goal, we have developed an application that allows deploying a minimal number of RSUs using two metaheuristics: genetic algorithms and simulated annealing. The results obtained are promising since we have been able to deploy less than 50% of the RSUs at intersections.

ملخص

لضمان نشر المعلومات بطريقة فعالة في شبكات VANET ، يوصى باختيار بنية تكون اتصالاتها من النوع V2I حيث تكون وحداتRSU في التقاطعات .وبما أن تكلفة هذه RSUs باهظة الثمن فإنه يتطلب تقليل عددها مع ضمان معايير توجيه مقبولة. للقيام بذلك قمنا بتطوير تطبيق يسمح بنشر عدد قليل من وحدات RSU باستخدام اثنين من :metaheuristics الخوارزميات الجينية ومحاكاة التلدين .النتائج التي تم الحصول عليها واعدة حيث تمكنا من نشر أقل من 50 ٪ من RSUs عند التقاطعات.